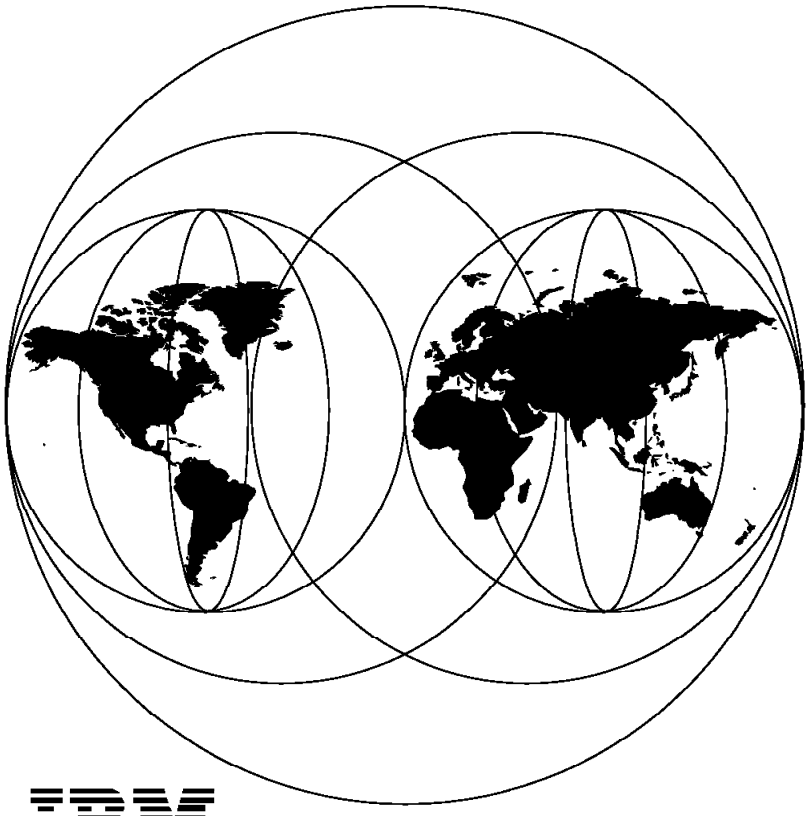
SG24-4815-01

Cool Title About the AS/400 and Internet

November 1996



IBM

International Technical Support Organization
Rochester Center

International Technical Support Organization

SG24-4815-01

Cool Title About the AS/400 and Internet

November 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special Notices” on page 297.

Second Edition (November 1996)

This edition applies to Version 3 Release 2 of the TCP/IP Connectivity Utilities/400, 5763-TC1 for use with Operating System/400, 5763-SS1.

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

Contents

Preface

This document is instrumental in the access and use of the Internet (or your own intranet) from your AS/400. It helps you understand how to use the functions and features available with V3R2 of OS/400 and the TCP/IP Connectivity Utilities/400, also known as the Internet Connection for AS/400. This document can also be used for V3R7 of OS/400 due to the functional equivalence with V3R2.

This redbook helps you get your AS/400 ′plugged in′ to the Internet with a discussion of Wide Area Network (WAN), Local Area Network (LAN) and both dial-out and dial-in SLIP connections.

We then will address some traditional TCP/IP applications like e-mail with MIME and POP3 enhancements, Telnet, FTP with new APIs to provide anonymous access, Gopher, and AS/400 security issues.

Then, we will detail the new ′cool′ side to the AS/400 with the World Wide Web (WWW) HTTP server as part of the Internet Connection for AS/400. Topics such as CGI programming, access to DB2/400 data via DB2WWW macros, and the 5250 Workstation Gateway that converts legacy 5250 applications to the language of the web (HTML) are addressed with implementation details and example programs.

All the source programs written in HTML, RPC, C, DB2WWW macro, plus a three day class to support the creation of this redbook are included on a single CD-ROM.

How This Redbook Is Organized

This redbook contains 302 pages. It is organized as follows:

- Chapter 1, “The Internet Overview”

  This chapter gives you an overview of the Internet and of Network Computing. There should be no surprises here for most people.

- Chapter 2, “Your AS/400 System and the Internet”

  This chapter starts to focus on the AS/400 system and its relationship to the Internet. Here we cover such issues as:

  - What options you have when connecting your AS/400 system to the Internet.
  - Once connected, what applications are available on the AS/400 system.
  - What to look for in an Internet service provider.
  - What hardware and software you will need on your AS/400 system to get online.

- Chapter 3, “Connecting to Your ISP”

  This chapter goes into more detail about how to configure your AS/400 system to connect to an ISP (Internet Service Provider). Two basic scenarios are discussed:

  - The AS/400 on a Local Area Network (LAN) connected to an ISP via an ISP provide router.
  - Dial-up connections using SLIP

- Chapter 4, “E-mail, SMTP, and POP3”

  OK, now that we are online with our AS/400 system, what can we do with the sending and receiving of electronic mail? This chapter answers this question and more.

- Chapter 5, “Telnet on the AS/400 System”

  Telnet (terminal emulation over a network) client and server issues are covered in this chapter.

- Chapter 6, “FTP on the AS/400 System”

  FTP (File Transfer Protocol) client and server are covered in this chapter, along with plenty of real scenarios of how to effectively use FTP with your AS/400 system on the Internet.

- Chapter 7, “Gopher on the AS/400 System”

  Gopher, a text based search and retrieval tool, client and server are covered in this chapter.

- Chapter 8, “Serving the World Wide Web from Your AS/400 System”

  The AS/400 system′s Internet Connection for AS/400 web server product is described in great detail. This chapter discusses what the HTTP (HyperText Transfer Protocol) server on the AS/400 can do, and then how to configure it. It also takes a look at security in relation to the HTTP server.

- Chapter 9, “Application Development Interfaces for World Wide Web”

  The Internet Connection for AS/400 provides a number of ways to access legacy data on your AS/400. This chapter addresses the following:

  – Common Gateway Interface programming allows you to pass parameters between the remote web client and your AS/400 HTTP server and application program.

  – The server-side image map support allows the AS/400 to branch to a different URL (Universal Resource Locator) based upon the location of the graphics cursor within an image when the mouse was clicked.

  – DB2WWW (which stands for DB2 World Wide Web) is a macro language that greatly simplifies the access and update of SQL based data on your AS/400 or in your network. This macro language is also known as net.data.

  – The 5250 to HTML gateway is a feature of the Internet Connection for AS/400 that dynamically converts the 5250 data stream into the language of the World Wide Web, HTML (HyperText Markup Language). This allows most of the worlds traditional 5250 applications to be now run over the Internet or intranet from a simple web client.

- Chapter 10, “Security and Audit on the Internet”

  This chapter looks at security issues with the Internet and your AS/400. It addresses security issues that cross all the different TCP/IP applications and looks at security from the system point of view. If you are interested in security for FTP, as an example, go to the FTP chapter.

- Appendix A, “Installing the ITSO Company Demo”

  This book comes with a working application and a three day class all written in HTML along with the source for the supporting AS/400-side of the application in save files. This chapter leads you through the steps to restore,

configure and run the many useful tools and sample applications provided on the electronic media that came with this book.

- Appendix B, “How Are OS/400 Users Counted for User-Based Pricing”

  This chapter describes how OS/400 licences are counted when you use TCP/IP functions. The rules are different between V3R1 and V3R6.

The Team That Wrote This Redbook

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

Heikki Arhippainen is a systems engineer in Finland. He has 14 years of experience in the S/38 and AS/400 field. He has worked at IBM for 6 years. His areas of expertise include communications, performance and communications programming.

Richard Halleen is an AS/400 Supportline Representative in Rochester. He has worked extensively with AS/400 TCP/IP for the past three years. He also has experience with all of AS/400 communications software. He has worked at IBM for 14 years, all of which has been in software service.

Lee Hargreaves is a Technical Support Specialist in the UK. He has 7 years of experience in AS/400 communications field all of it with IBM. His areas of expertise include TCP/IP and APPC communications. He has written extensively on TCP/IP SNMP.

James Hudlow is a I/T Services Specialist in Atlanta Georgia. He has 8 years of experience in AS/400 technical support. His areas of expertise include AS/400 data migration (GIGMIG) and Client Access/400.

Bernd Lindner is a IT/SE Specialist in Germany. He has 25 years of experience in IBM S/3, S/38 and AS/400 field, all it with IBM. His areas of expertise include communications, communications programming and systems management. He has written extensively on CGI-BIN programming for this redbook.

Brian R. Smith is a Advisory International Technical Support Specialist at the International Technical Support Organization, Rochester Center. He writes extensively and teaches IBM classes worldwide on all areas of AS/400 communications, specializing in TCP/IP. Brian has worked his entire 12 year career with IBM Rochester first with the System/38 and later with the design, development, test and support of the AS/400.

S W Wu is a Technical Sales Specialist in Taiwan. He has 7 years of experience in the AS/400 field. He holds a Bachelor degree from Tsing-Hua University. His areas of expertise include programming and the Internet. He has written extensively on DB2WWW and HTML Gateway for this redbook.

The authors of the first edition of this document were:

Peter Eibak
IBM Denmark

Detlef Fallisch
IBM Germany

John Parker
IBM USA (Iowa, in fact)

Jener Takeshi Sato
ITEC Brazil

Rika Schwenkenbecher
IBM Australia

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

Stefan Imhof
IBM Switzerland

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. The Internet Overview

This redbook is intended to give you directions to connect an AS/400 system to the Internet. It gives you the necessary information to install, configure, and run the Internet-related AS/400 applications as well as what to look for in a service provider or type of connection and so on.

This book, the same as other IBM Redbooks, mentions related areas such as IBM offerings and AS/400 solutions.

As you may have seen, most product or solutions providers are offering many kinds of Internet services. As well as an Internet connection for the AS/400 system, IBM offers:

- Client and server software
- Firewalls
- IBM networking services
- IBM information super highway solutions
- IBM Global Network (IGN)

In this chapter, we talk about:

Topic	Please see
What is the Internet	See 1.1, “What is the Internet” for more information.
History of the Internet	See 1.2, “History of Internet” on page 2 for more information.
Network Computing	See 1.3, “Network Computing” on page 3 for more information.
The Application Layer	See 1.5, “The Application Layer” on page 5 for more information.

1.1 What is the Internet

The Internet is *the* network of networks. It is an interconnection of many smaller networks forming a larger network we call the Internet. As the number of hosts and other networks connecting to the Internet grow so does the Internet.

Since the Internet’s conception, applications and protocols have been developed to utilize it. The popularity of applications such as FTP and E-mail caused a meteoric rise in the number of users connecting to the Internet. More recently, however, the World Wide Web is the area of usage showing the most rapid growth.

The original backbone of the Internet was provided by the U.S. Government, Science institutes, and Universities as a means of sharing research data. As usage has outgrown the original infrastructure, it has been continually upgraded by communications companies funded by the subscriptions of the end users.

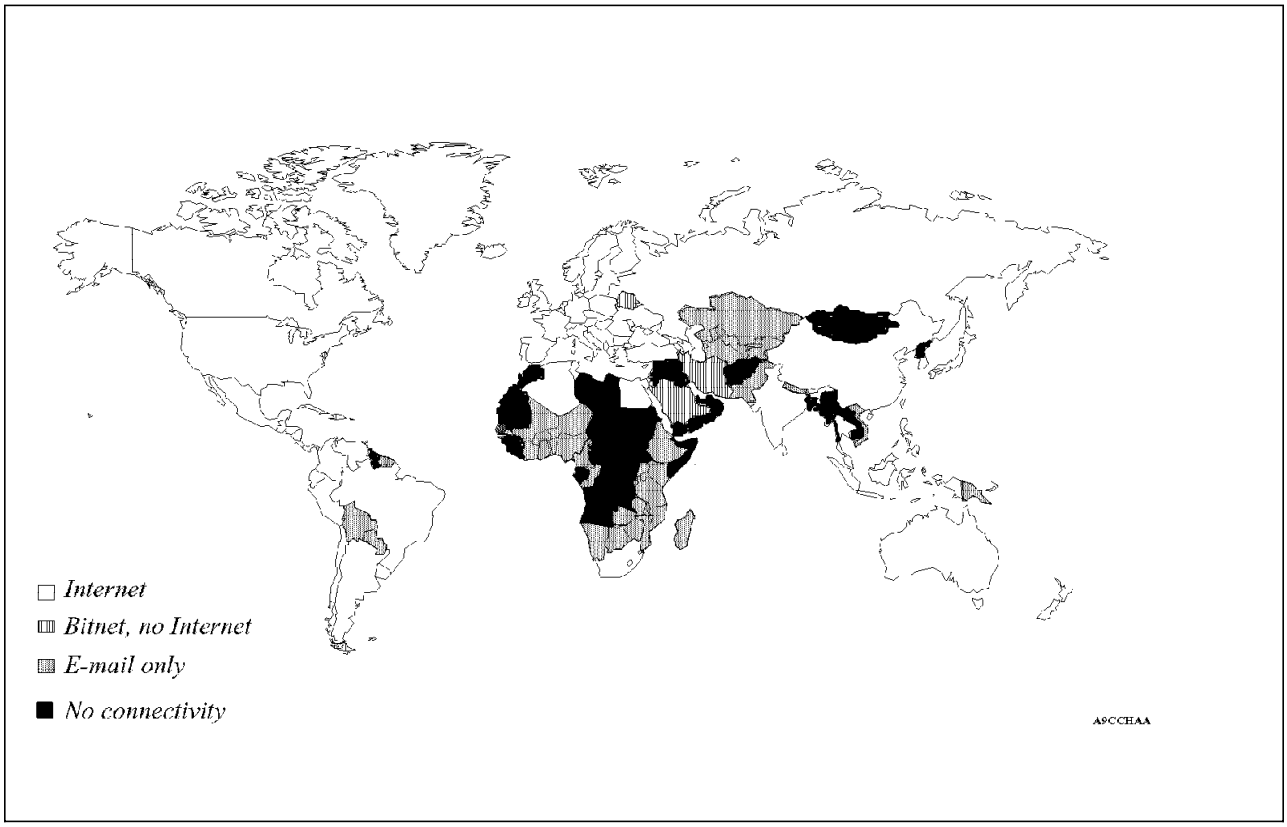


Figure 1. Internet World Map

1.2 History of Internet

The Internet was born in the late 1960s as an effort to create a United States Department of Defense network called ARPANET. The main purpose for it was to connect defense contractors and universities that were working on projects for the Department of Defense.

In 1973, the ARPANET established connections to England and Norway, and in 1982, the TCP/IP became the protocol suite for ARPANET. This leads to one of the first references to an ″Internet″ of connected networks.

In 1983, the University of Wisconsin developed the Name Server that was introduced in 1984.

In 1986, NFSNET was created with a backbone speed of 56 Kbps and upgraded up to T1 (1.544 Mbps) in 1989.

In 1989, the AS/400 system started supporting TCP/IP on Release 2.0 of OS/400. This was even before we knew that this was version 1!

Gopher was introduced by the University of Minnesota in 1991. WWW (World Wide Web) was released by CERN in 1992 adding to established and popular applications such as Telnet, E-mail, and FTP.

Today, we have an estimated 4 000 000 hosts connected to the Internet and many more connected to internal ″intranets″. Network Providers have upgraded

the speeds of their backbones to T1, T2, T3, and so on to cope with the rising demand for bandwidth.

Now, with the Internet on the verge of becoming a huge commercial marketplace, the challenge is to create a suitable environment for secure electronic commerce.

Here are some interesting Internet facts:

- USA Corporations establish a new site on the Internet every 12 minutes - Internet Society.
- By early 1996, there were 170 000 commercial sites on the Internet worldwide - InterNIC.
- USA Shoppers purchased 436 million U.S. dollars worth of goods and services over the world wide web in 1995 - ActivMedia.
- In 1996, 81% of the top 200 U.K. companies viewed the Internet as a business opportunity, with more than one-third already having an Internet site - Barclays Bank.
- In Asia, Europe, and the U.S., about 17.6 million people are expected to be connected to the Internet by the end of 1996, up from 10.6 million in 1995 - Forrester Research.
- 50 years after the invention of the telephone, world residents had made a total of 38 million phone calls. Today on the Internet, around 38 million page visits occur every 24 hours - SRI International.

1.3 Network Computing

The latest stage in the computing industry's evolution is the Client/Server model in which end users (clients) are able to use the services of multiple host systems (servers). These customers require highly effective products, services, and solutions to help integrate all of the legacy data from distributed systems. This takes us to the Network Computing (NC) model.

An example of the Network Computing model can be:

A single AS/400 system accessing vast amounts of data, stored on many different systems across a large network as if it were stored on a single entity. End users perceive information, applications, and services as being provided by the network rather than by a computer system. IBM is committed to Network Computing on the AS/400 system, a system that has the advantages of an integrated database, integrated security, and an object-based structure.

The AS/400 users can utilize the latest technology, reduce or eliminate many geographic barriers, exploit the Internet to their companies, and enable new ways of doing business.

Internet Connection for AS/400, which is an integrated package included with OS/400, helps customers conduct business on the Internet. It includes:

- World Wide Web Hypertext Transfer Protocol (HTTP) Server
- Workstation Gateway (WSG) Server
- DB2WWW
- Logging of World Wide Web Server access

TCP/IP has also been enhanced to provide:

- FTP support across all of the Integrated File System (IFS).
- FTP authentication exits points (Anonymous FTP).
- Asynchronous communications support in the form of SLIP.

Reference: URL http://www.as400.ibm.com

1.4 Why Attach an AS/400 System to the Internet?

Why not? If you are seriously considering putting a server onto the Internet, then there are many distinct advantages in choosing an AS/400 system. If you have concerns about using an AS/400 system in this role, then ask yourself whether the reason for your concern, such as security, is associated with the Internet itself rather than a specific system.

Consider the advantages of using an AS/400 system:

Communications An AS/400 system has exceptional communication capabilities. LAN, WAN, and dial-up capability are all part and parcel of OS/400 providing a choice of connection methods using TCP/IP.

Scaleability The common features present in OS/400 across the entire model range make it easy to accommodate future growth.

Functionality The AS/400 system provides the popular TCP/IP applications such as Telnet, FTP, E-mail, WWW Server, and so on.

Software cost Almost all the AS/400 TCP/IP suite of applications comes free of charge when you order OS/400.

Training If you are already running an AS/400 operation, then the training required to attach your AS/400 system successfully to the Internet is minimal.

Security The AS/400 system has extremely good, native, security features. These features alone can prevent many Internet-related security problems that afflict other systems. With user exit programs, you can further enhance the security for FTP and Workstation Gateway Servers.

Legacy Data Perhaps the main reason to consider an AS/400 system as an Internet server is when the information you want to serve is already resident on an AS/400 system. The new Internet Connection for AS/400 applications provide several methods of accessing your legacy data through the Internet. By using the Workstation Gateway Server, most of your existing ″green-screen″ applications are Internet enabled now. With DB2WWW, you can access your legacy data from WWW pages. With CGI programs, you can build new applications to be used from WWW pages.

1.5 The Application Layer

The applications layer for Internet includes all programs that use TCP/IP and C-Sockets to communicate over the Internet. These applications are:

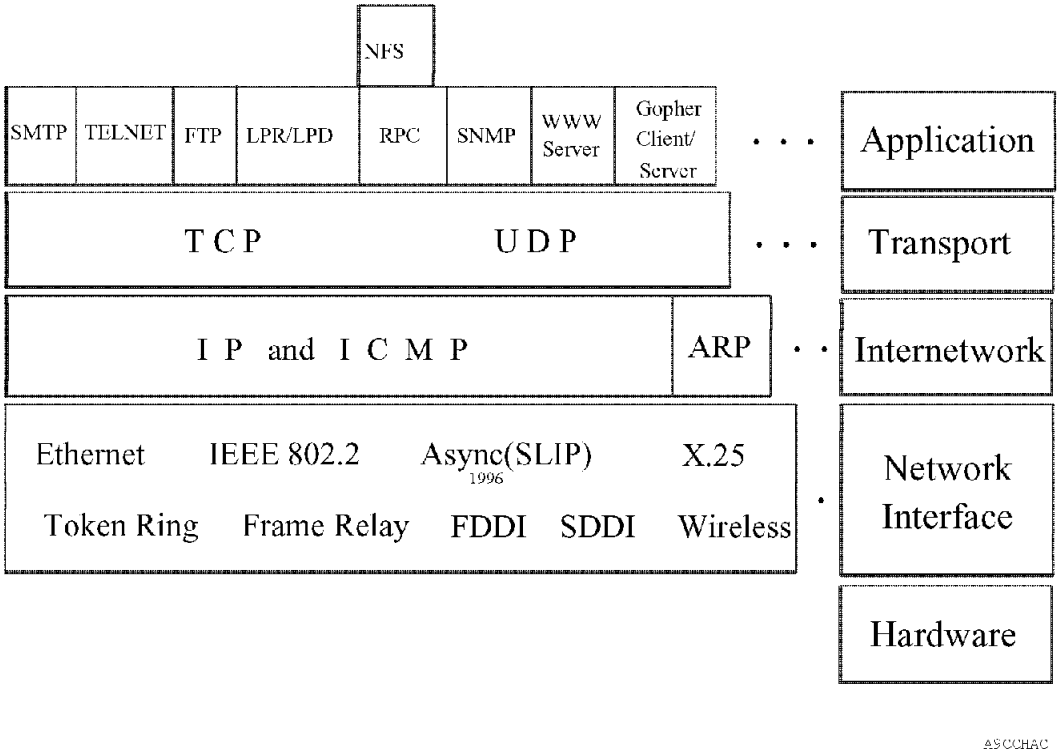


Figure 2. TCP/IP Applications Layer

OS/400 and the free Licensed Program TCP/IP Connectivity Utilities/400 provide support for the following applications and functions, many of which we explain in detail in this Redbook.

Application Description

Telnet Client/Server

The AS/400 Telnet client allows an AS/400 5250 terminal to sign on and run remote applications on any host across the Internet using a TN5250, TN3270, VT220, or VT100. The AS/400 Telnet server allows just the opposite; any remote terminal connected to another host can sign on and run an AS/400 5250 application. See Chapter 5, “Telnet on the AS/400 System” on page 91 for more information.

FTP (File Transfer Protocol) Client/Server

The AS/400 FTP client allows you to ″be in the drivers seat″ as you connect to a remote host to either get files from your AS/400 system or put files to that remote host. The AS/400 FTP server waits for connections from other remote FTP clients who are also either putting files to your AS/400 system or getting files from it. See Chapter 6, “FTP on the AS/400 System” on page 105 for more information.

LPD (Line Printer Daemon)/LPR (Line Printer Requester)

With LPD, the AS/400 system can be a printer server in a TCP/IP network, and LPR allows the AS/400 system to request a file to be

printed at another host. See *TCP/IP Configuration and Reference - Version 3*, SC41-3420-00, for more information about LPR and LPD on the AS/400 system.

Gopher Client/Server

The Gopher Server put the AS/400 system as a data server on the Internet where any Gopher client can access the information through a Gopher interface. Also, the Gopher client allows the AS/400 system to access other Gopher servers on the Internet. See Chapter 7, “Gopher on the AS/400 System” on page 149 for more information.

HTTP Server (WWW)

By using the AS/400 Hypertext Transfer Protocol (HTTP) Server, it is possible to serve Hypertext Markup Language (HTML) pages to Web browsers across the Internet. This positions the AS/400 system as a server in the fastest growing area of the Internet. The HTTP server can also act as a gateway between the client and the AS/400 database, using the CGI interface or by utilizing the precoded CGI programs in DB2WWW. See Chapter 8, “Serving the World Wide Web from Your AS/400 System” on page 159, 9.1, “Common Gateway Interface Programs” on page 191, 9.4, “Server Side Image map Support” on page 260, and 9.2, “DB2 World Wide Web Connection” on page 216 for more information.

Workstation Gateway (WSG) Server

This is a TCP/IP application that transforms AS/400 5250 data streams to HTML for dynamic display on Web browsers. This enables you to run AS/400 applications from any workstation that has a Web browser. See 9.3, “5250 HTML Work Station Gateway” on page 233 for more information.

SMTP & POP3

The AS/400 system has been able to send E-mail over the Internet using SMTP since Release 2.0. The Post Office Protocol (POP3) Mail Server enhances that ability. It provides a store-and-forward mechanism using electronic mailboxes on the AS/400 system from which clients can retrieve mail. See Chapter 4, “E-mail, SMTP, and POP3” on page 57 for more information.

SLIP Serial Line Internet Protocol (SLIP) is a popular method of connecting two systems over a pair of modems that are connected over a telephone line. One use of this is to provide dial-in connections to your AS/400 HTTP Server. See 3.2, “SLIP Dial-Up Support on the AS/400” on page 28 for more information.

Chapter 2. Your AS/400 System and the Internet

This chapter describes all of the steps that you need to follow to get your AS/400 system connected to the Internet. And then, once connected, describes the functionality that you have available on your AS/400 system.

These steps include:

- Getting your AS/400 system connected to the Internet: See 2.1, “Getting Your AS/400 System Connected to the Internet” for more information.
- What TCP/IP applications are available on the AS/400 system: See 2.2, “The TCP/IP Applications Available on the AS/400 System” on page 14 for more information.
- Choosing an ISP: See 2.3, “Choosing an ISP” on page 15 for more information.
- What you need on the AS/400 system: See 2.4, “What You Need on AS/400 System” on page 18 for more information.

2.1 Getting Your AS/400 System Connected to the Internet

The first thing you have to do is choose the type of connection you are going to use to connect your AS/400 system to the Internet.

This topic gives a brief description in the following communication interfaces:

1. WAN (Wide Area Network): Please see 2.1.1, “WAN (Wide Area Network)” for more information.
2. LAN-to-LAN (Local Area Network): Please see 2.1.2, “LAN-to-LAN (Local Area Network)” on page 10 for more information.
3. Dial-up Connection: Please see 2.1.3, “Remote or Dial-Up” on page 12 for more information.

For each protocol, the following items are covered:

1. Cost effectiveness
2. Performance as determined by workload and speed lines
3. ISP constraints
4. Popularity of the connection method
5. Reliability
6. Security comparisons
7. Time frames

2.1.1 WAN (Wide Area Network)

We have two wide area network connectivity options on the AS/400 system, X.25 and Frame Relay.

1. X.25: Please see 2.1.1.1, “X.25” on page 8 for more information.
2. Frame Relay: Please see 2.1.1.2, “Frame Relay” on page 9 for more information.

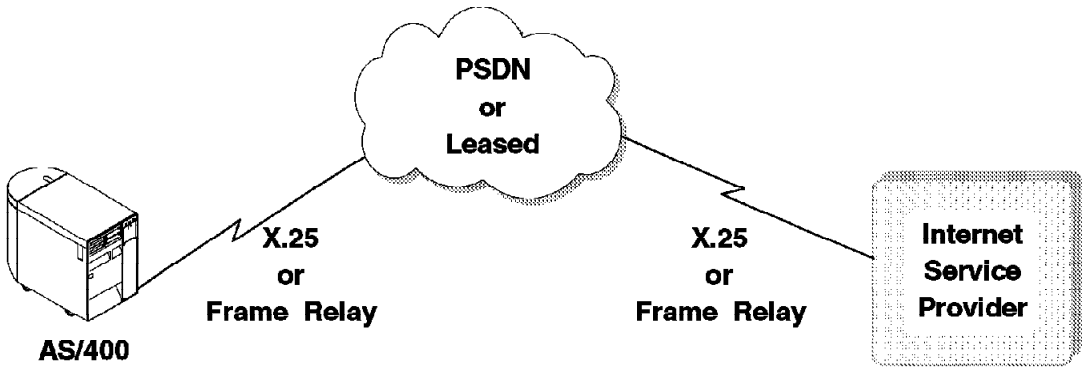


Figure 3. Example of a WAN Connection for the AS/400 System

2.1.1.1 X.25

The following list explains some considerations about this type of connection:

1. Cost effectiveness:

   On the AS/400 system, you need a synchronous line; the protocol is provided within the operating system. Depending on whether or not your provider has X.25 access through a direct connection or through a Packet Switch Network, you must pay for a leased or dial line.

   The X.25 strengths:

   - Both public and private options.
   - Packet switching approach allows statistical access to available bandwidth.
   - Efficient for bursty traffic.
   - Provide virtual circuits for any-to-any connections.
   - Error detection and correction.

   The X.25 weakness:

   - The X.25 protocol requires error correction generally limiting access speed to 256 Kbps, although a few products support T1 speeds.
   - Extensive protocol processing increases the price of switches and the price of services as well.
   - The number of ISPs that provide X.25 connection are limited.

2. Performance as determined by workload and line speeds:

   As we have seen, this protocol increases the amount of processing required for error correction and protocol processing which limits the speed. On the AS/400 system, the X.25 link can be configured up to 2048 Kbps depending on which adapter you are using (V.24, V.35, or ISDN).

3. ISP constraints (communications interfaces supported):

   On this configuration, you can have two different situations:

   - Direct connection to the service provider:

Some providers may have an X.25 port that can be used for a direct connection. Once the AS/400 system can be a DCE/DTE on a X.25 network, you are able to connect your AS/400 system directly to the provider's X.25 port. This connection can be done through a leased or dial line.

- Connection through a packet switch network:

  If the provider only has X.25 access through a packet switch data network, then you have to sign up for a leased line and for X.25 access. Be careful if you are planning to send or receive large amounts of data as normally the packet switch providers charge by the packet. Therefore, it can be expensive.

4. Popularity of the connection method:

   Depending on where you are, (in which country or state), you may not have an X.25 facility. Or - most likely, you can have access to an X.25 PSDN but cannot find an ISP that will provide Internet access.

5. Reliability (dial-up versus lease-line link):

   If you are going to have services on the Internet and need high reliability, it is better to have a leased line instead of a dial-up line. The dial-up line is best if you do not need high reliability and a speed higher than 28.8 Kbps.

6. Security comparisons:

   For X.25 connections, once your AS/400 system is connected to the ISP, you must set up the highest level of security needed because anyone can reach your machine.

7. Time frames:

   The X.25 connection is supported by the AS/400 system and it is integrated in the operating system. In V3R1, the X.25 supports TCP/IP connections over a switched virtual circuit (SVC) and a permanent virtual circuit (PVC).

2.1.1.2 Frame Relay

The frame relay networks are made up of frame relay access equipment, frame relay switching, and public frame relay services.

- Frame relay access equipment is basically a host, router, or bridges.
- Frame relay switching equipment is a device that is responsible for transporting and routing. It can be an E1/T1 multiplexer, packet switch, and so on. This equipment can also be used to build a private frame relay network.
- Public service providers offer frame relay access by deploying frame relay switch equipment.

The following list explains some considerations about this type of connection:

1. Cost effectiveness:

   On the AS/400 system, you need a synchronous line; the protocol is provided within the operating system. Depending on whether or not your provider has direct access or access through a public network service, you have to consider how expensive this access may be through a public service.

2. Performance as determined by workload and line speeds:

You have to consider the amount of data that is flowing through your connection to configure the correct line speed.

Generally a 56/64 Kbps or E1/T1 is used.

3. ISP constraints (communications interfaces supported):

   On this configuration, you can have two different situations:

   - Direct connection to the service provider:

     Some providers may have a private frame relay network so you can be attached directly to their network through statistical multiplexers. If so, you are able to connect your AS/400 system directly to the provider.

   - Connection through a public service provider:

     If the providers only have a frame relay access through a public service, then you must contact your local public service provider for a frame relay access.

4. Popularity of the connection method:

   At this time, you may have some problems with this type of connection because very few ISPs offer frame relay as a connection media.

5. Reliability (dial-up versus lease-line link):

   If you are going to have services on the Internet and need high reliability, it is better to have a leased link instead of a dial-up link. The dial-up link is best for you if you do not need high reliability and a speed of more than 28.8 Kbps.

6. Security comparisons:

   Once you are connected to the Internet, you must be aware of security items. If possible, you can set up a firewall in another machine. Please see Chapter 10, “Security and Audit on the Internet” on page 267 for more information.

7. Time frames:

   The frame relay connection is supported on the AS/400 system and is integrated into the operating system.

2.1.2 LAN-to-LAN (Local Area Network)

This sub-category applies to such topology as where both or one side has a LAN. As we are talking about an Internet that involves the TCP/IP protocol, the SLIP or PPP connection is the standard for remote connections.

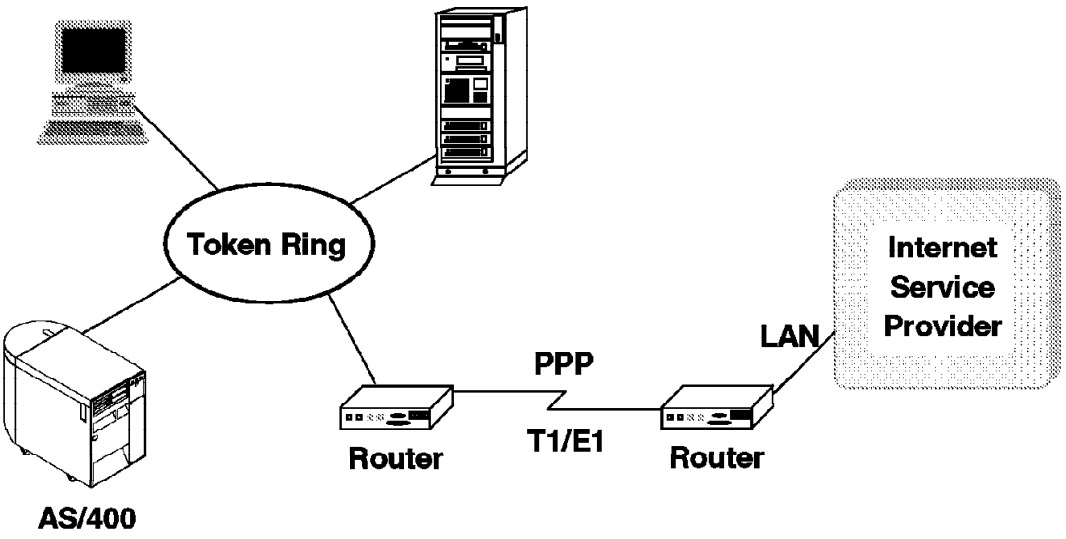


Figure 4. Example of a LAN-to-LAN Connection for AS/400 System

In this environment, you may have your AS/400 system connected to your local area network (LAN). Your LAN can be an Ethernet, token-ring, wireless, or any other LLC that supports TCP/IP.

To get your LAN connected to an ISP that may or may not have a LAN, most of the ISPs offer the installation and configuration of the routers with no fee.

If they do not offer this facility, there are more details on how to configure this environment in the next chapter. See 3.1.1, “Setup of IP Router” on page 23.

The following list contains some considerations about this type of connection:

1. Cost effectiveness:

   For this environment, you have to consider the following:

   - Acquire a router that supports PPP, IP routing, and, of course, the type of LAN interface you have (Ethernet or token-ring).
   - Type of link to connect to the provider: leased or dialed. Generally, the routers have an interface for V.24, V.35, and RS-442.

2. Another point to consider is that not only your AS/400 system can use the local IP router to gain direct access to the Internet, but any other system directly attached or in the same subnet as your AS/400 system.

   You must be certain, however, as you are negotiating with the ISP whether you are purchasing just a single IP address for your AS/400 system or arranging for an entire subnet for all of your systems.

3. Performance as determined by workload and line speeds:

   Generally, you can pay for the speeds you want from 56 Kbps to E1/T1/J1 speeds.

4. ISP constraints (communications interfaces supported):

   This type of connection can be made only with a direct connection. You have to contact a provider that offers this type of service.

5. Popularity of the connection method:

   Most of the providers offer a direct connection through SLIP/PPP to their location. They may have many SLIP/PPP ports on their host or sometimes they use a set of routers. Therefore, it does not matter if you are using a router or a host port to connect through SLIP/PPP.

6. Reliability (dial-up versus lease-line link):

   If you are using a router on a PPP connection, then it is better to use a leased line because the routers cannot be configured to automatically re-dial. If you need high reliability, use a leased line.

7. Security comparisons:

   When you use the LAN-to-LAN connection through IP routers and you have more hosts connected to your LAN through a TCP/IP protocol, you must configure the highest level of security because your LAN is open to anyone.

   If you only have your AS/400 system connected to the router, and your LAN is connected to another LAN adapter of the AS/400 system, you can limit the access of non-welcome visitors by configuring the IP-forwarding parameter on the AS/400 system's TCP/IP to NO. This way, everyone that wants to go through your private LAN must first sign on to the AS/400 system.

8. Time frame:

   This is available because you have a LAN Adapter on your AS/400 system.

2.1.2.1 Other Types of Connections

There are others types of connections also available on the AS/400 system, but we are not going to talk about them because they are not widely available.

The connections are:

- SDDI
- FDDI
- ISDN
- Wireless

2.1.3 Remote or Dial-Up

The third type of connection is for a remote or dial-up connection; the main characteristic is that you do not need a full-time connection.

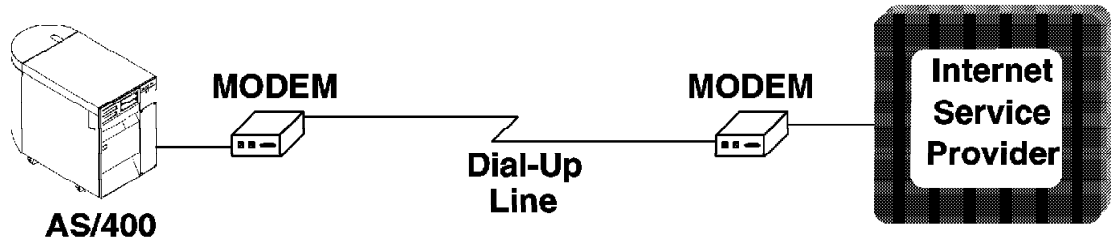


Figure 5. Example of a Dial-Up Connection for AS/400 System

This connection is normally used when you need a cheap and low availability connection to your AS/400 system. The following list contains some considerations about this type of connection:

1. Cost effectiveness:

If you do not need a leased connection with a high speed such as T1/E1, you can use a SLIP connection. This switched solution is probably much cheaper than the lease line solution, but does not have all of the advantages of the lease line either.

You can use a communications port to support the SLIP protocol. This, again, should be inexpensive for most customers as most AS/400 systems come with an ECS line that can be used for this purpose.

- The SLIP connection can use the following adapter types:
  - 2609 - Two-line EIA 232/V.24 adapter
  - 2612 - One-line EIA 232/V.24 adapter
  - 6152 - Two-line EIA 232/V.24 adapter

  The present set of IOPs that SLIP can run on are the 2623 and the MFIOPs from the D02, D25, all E models, all F models, and all Advanced Series models. There are some unsupported combinations. For details, see the *AS/400 Advanced Series Handbook*.
- A variety of well-known (Hayes compatible) ASCII modems are supported.

Using the current stage, Async IOP/IOA speeds of 1200, 2400, 4800, 9600, 14400, and 19.2 Kbps are supported. ISPs support of 19.2 Kbps is rare, so the most common highest denominator is 14 400 bps. A 28.8 Kbps modem can be used, but the effective throughput is gated by the maximum DTE speed of 19.2 Kbps. The AS/400 system uses hardware flow control to ensure that data over-runs and under-runs do not occur due to error re-transmission and data compression down at the modem level.

2. ISP constraints (communications interfaces supported):

   The SLIP and PPP connections are widely available for ISP. These are the most common types of connection.

3. Popularity of the connection method:

   This is the most popular method to connect to an ISP. Generally, these two protocols are supported by the ISP as a dial-up connection.

4. Reliability:

   The reliability of a dial-up connection is directly dependent on the quality of the line. If you do not need high reliability for your connection, then this should be the best and the cheapest one.

5. Security comparisons:

   Once you have a dial-up connection, if you do not start a connection, you can say that your system is secure. But, once you are connected, you need the same levels of security as any other type of connection. This happens because your host can be accessed by anyone in the same network. Therefore, you must be aware of all types of protections you can use on your system.

6. Time frame:

   This is integrated into the operating system.

2.2 The TCP/IP Applications Available on the AS/400 System

There are several ways to connect your AS/400 system to the Internet. These are covered in more details later in this book. As you know, the Internet World is based on the client/server model, so when you connect your AS/400 system to the Internet, it is a client requesting services to a host already connected to the Internet.

As in the local TCP/IP Network, the AS/400 system can be used as a client, server, or both, depending on what kind of applications you need through the Internet.

Application	Client	Client Description
	Server	**Server Description**
Telnet	Client	Allows you to connect to other hosts with a terminal emulation, such as VT100, VT220, TN5250, or TN3270. See 5.1, “The Telnet Client on the AS/400 System” on page 91 for more information.
	Server	Allows others hosts to access the AS/400 system through Internet and have a Signon display. See 5.2, “The Telnet Server on the AS/400 System” on page 98 for more information.
FTP	Client	Allows you to put and get data into a remote file system. See 6.3, “FTP Client on the AS/400 System” on page 119 for more information.
	Server	Make the AS/400 system′s file system available to be accessed by other hosts to get and put data. See 6.2, “FTP Server on the AS/400 System” on page 116 for more information.
Printer	Client (LPR)	Allows you to request your file to be printed at a remote host through the Internet. See *TCP/IP Configuration and Reference*, SC41-3420.
	Server (LPD)	Allows the AS/400 system to be a printer server, which means that any host can request a file to be printed on an AS/400 printer. See *TCP/IP Configuration and Reference*, SC41-3420.
NFS	Client	Not Supported
	Server	The Network File System allows the AS/400 system to be a file server on the Internet. See *The IBM AS/400 as a TCP/IP Network File Server*, GG24-4092.
Gopher	Client	Allows you to access the Gopher servers in the Internet. See 7.2, “AS/400 Gopher Client” on page 152 for more information.
	Server	Allows the AS/400 system to be a Gopher server, which means that any host can access the AS/400 databases, folders, and so on through a text-oriented interface. See 7.1, “The AS/400 Gopher Server” on page 151 for more information.
HTML 5250 Gateway	Client	Does not make sense!
	Server	Enable any Web browser to access most of the AS/400 system′s applications without any modifications. See 9.3, “5250 HTML Work Station Gateway” on page 233 for more information.
HTTP (WWW)	Client	Not available
	Server	The Web server or HTTP (HyperText Transport Protocol) Server allows the AS/400 system to be a WWW server. This allows the AS/400 to serve HTML documents from either the Integrated File System (IFS) or out of QSYS.LIB. See Chapter 8, “Serving the World Wide Web from Your AS/400 System” on page 159 for more information.

2.3 Choosing an ISP

When you contact an ISP, you are really buying access to the Internet. At this point, you should be driven by your intended use. If you are just connecting to have personal access, you should look for minimum cost. But, if this connection is intended for your company′s use, you probably choose differently. The next step you have to consider is: Which ISP should I contract? For an answer to this question, you should cover the following options:

1. Connectivity options
2. Cost
3. Performance
4. Functionality available
5. Service availability
6. Growth options
7. Security options

And, the answer to the question of which ISP to contact is obvious. Contact them *all* asking the questions raised in this section. Choose the ISP that gives you the best service for the money. Simple, no?

Here is a simple dialog that could be the start of your quest to find the best ISP in your area.

You: Hi. I have an AS/400 system and some money and I want to set this system up to be a permanent server on the Internet.

ISP: An AS/400 system? What is an AS/400 system?

You: The most popular midrange business computer in the world. I have money.

ISP: OK, money. I can sell you a low cost 56Kbps line using PPP that is tied directly to the Internet backbone through MCI with a T1. We can also price a T1 (1.544 Mbps) also, or anything between.

You: Well, my AS/400 system does not have PPP yet. But, it does have:

FDDI and SDDI
Wireless
X.25
Frame Relay
Token-Ring
Ethernet...

ISP: OK, stop. Ethernet is good. We can drop in an IP router on your Ethernet LAN. We will configure the IP router and its PPP connection back to our place over a leased 56 Kbps line from MCI. We will get you a fixed IP address, domain, and provide for you a domain name server. All you have to do is to configure your, ahh, 400 thing, to use our IP router and domain name server. I can fax you some sample configuration options including the final cost to you, both initial and monthly.

You: Cool.

Yes, this is somewhat in jest, but shows that a majority of the responsibility lies with the ISP. You are paying them to provide a service. The ISP that provides the best service for the lowest cost should win your business. In the next

section, we discuss some of the important questions to ask each and every ISP in your area.

An updated list of ISPs is kept on the Internet at `http://www.primus.com/providers/`.

2.3.1 Connectivity Options

Your have to consider if the ISP has the type of connection you need or want to use.

Here is where the limitations of connectivity to certain ISPs come in. Some ISPs do not have SLIP access, for example.

If the ISP does not have a SLIP access (only PPP), then your AS/400 system is restricted to a LAN-to-LAN connection using a PPP/Router connection, or a WAN connection through X.25, Frame Relay, or any other WAN access. It is better to verify which type of connection is better in cost/performance for your purpose.

2.3.2 Cost

Depending on the type of connection you are going to use, you have two options:

1. The AS/400 hardware and software costs (a one-time cost):

   On the AS/400 system, all of the software to connect to the Internet is *free* and, depending on the type of connection you intend to have, you must purchase a LAN adapter, Synchronous Serial adapter, and so on.

2. Cost connecting to the ISP (primarily running cost):

   Connecting to the ISP, the primarily running cost might be:

   - Line cost
   - ISP access cost

The first section is covered previously in connectivity options. The second is invariably tied to the other parameters mentioned such as the more bandwidth needed, the more expensive it is.

2.3.2.1 A Cost Scenario

As we explained before, for cost considerations, you have to look for which setup you really need and also consider the following items:

1. What is the one-time charge?
2. How much it costs per month?
3. If the ISP charges per time, get the ISPs unit for charge (hour, minute, second).
4. Get with your local network provider to determine the cost for the type and line speed you need.
5. For dial-up lines, check if you can have cheaper charges by connecting during off-peak hours.

2.3.3 Performance

For performance considerations, you need to verify with the ISP the following points:

- Network Topology: This is one of the most important criteria to consider when choosing a provider. Looking closely at the network topology, you can understand about the vulnerability, capacity, and the most important, how well the provider understands network engineering.
- Network Line Speeds: This is very important to give you an idea about which is the highest speed that you can ask for because it does not matter if you are connected to a T-3 node if there is a 56 Kbps link between you and your destination.
- External Network Links: If your ISP has a single external link, then this could be a potential point of failure.
- High-Speed Backbone: Verify if the speed the ISP is claiming if their backbone is available or is planned.
- There are many others considerations that should be considered for performance, but we are not going any further on this topic.

2.3.4 Functionality Available

The Internet Service Providers basically act in the following areas:

1. Information Service
2. Internet Service

2.3.4.1 Information Service

For this service, the client logs on to the ISPs server and usually performs limited and censored Internet functionality. The customer is cautioned to consider whether the AS/400 system is able to Telnet, FTP, and so on into the ISP easily for this.

2.3.4.2 Internet Service

This is where the ISP only provides IP routing to the rest of the Internet. However, you have to consider whether name serving is needed, for example.

2.3.5 Service Availability

You can check with other users for that ISP, the level of satisfaction they have, and also the reliability of the ISPs services, such as a line to an external connection, number of modems available, and so on.

Good questions to ask of the ISP are:

- How fast is your connection to the Internet?
- Does your connection to the Internet have a backup path in case the primary goes down?
- What is the current average and peak traffic volumes on your connection to the Internet? Use this answer to compare to other ISP answers.
- Describe your physical connection to the Internet.
- In the last year, what are your up-time statistics?

2.3.6 Support Structure

This focuses on timings of connection and reliability. The three major points that should be checked are:

1. Does the ISP have a backup line available? Does it have 24-hour availability?
2. Does the ISP have a real support structure, or just one person that attends to all of the customers?
3. If they have a support structure, ask them if they have a guarantee to respond or solve your problems within a certain time frame.

2.3.7 Growth Options

Does the ISP intend to offer PPP in the future if they do not offer it right now? These and other questions regarding access options should be asked when contacting an ISP. You have to keep in mind that the ISP supports an increased number of companies accessing the Internet through their backbone.

Also, verify if you can grow your environment without any restrictions.

2.3.8 Security Options

Does the ISP have firewalls available? Can you be sure that your company secrets are not tapped at the ISP side? Do they protect your IP addresses from being abused?

Verify if the ISP has firewalls, or something else to control or restrict the access to your network.

2.4 What You Need on AS/400 System

This section covers the software and hardware requirements on the AS/400 system.

2.4.1 Basic Software Requirements

These requirements are based on the premise that the AS/400 system is at V3R1 or later with TCP/IP bundled in.

To connect the AS/400 system to the Internet, you need the TCP/IP protocol. The AS/400 system has implemented TCP/IP since V1R2. With V3R0M5, it was *free* (no charge), and with V3R1 it is integrated into OS/400.

You can go through the same configuration steps you use to set up a local TCP/IP network (see *TCP/IP Configuration and Reference*, SC41-3420, and *TCP/IP Fast Path Setup*, SC41-3430), to connect to the Internet. There are additional steps to set up an internet connection that are described in more details in Chapter 3, “Connecting to Your ISP” on page 23.

Additional software, such as:

- Server code products needed (for example, Gopher, HTTPD)
- Client code products needed (for example, Web Browsers)
- Productivity tools needed (for example, HTML Editors)

These are covered in more detail in the following chapters.

2.4.1.1 TCP/IP on the AS/400 System

The TCP/IP protocol has been available on the AS/400 system since V1R2 as a separately-priced licensed program product (LPP) called the TCP/IP Connectivity Utilities/400. But, it was not until V3R1 that it was really integrated into the operating system (OS/400). At V3R1, the TCP/IP protocol stack and some functions were integrated into OS/400. The product, TCP/IP Connectivity Utilities/400, still exists and includes all of the applications such as Telnet, FTP, and so on, but is a free LPP that comes when you purchase OS/400.

Also on V3R1, we have integrated the C-Sockets interface into OS/400, so client/server applications can be written to work through the Internet or even in a local TCP/IP network. Most of the TCP/IP applications have been rewritten using a C-Sockets Interface on V3R1.

V3 Customers - Excluding V3R05: All of the integrated functions of TCP/IP are on V3R1 and the following list contains some of the highlights:

1. *Free* (no charge).
2. Many of the applications were rewritten from Pascal to ILE C/400.
3. Performance is comparatively better. It is up to eight times faster.

V2 and V3R05 Customers - Excluding V2R1: For those customer that have not upgraded to V3R1 and intend to install TCP/IP on their machines, it is probably cheaper if they go to V3R1 instead of purchasing the TCP/IP product for V2.

Table 1 (Page 1 of 2). Cross-Reference: OS/400 Licensed Program x Function

OS/400 Licensed Program	**Function**
OS/400 (5763-SS1)	TCP/IP protocol stack
	Configuration support for TCP/IP protocol stack
	Configuration support for SLIP **1**
	NETSTAT command
	PING command
	SNMP agent
	Configuration support for SNMP agent
	APPC over TCP/IP
	TCP/IP over AnyNet using sockets API
	C sockets API

Table 1 (Page 2 of 2). Cross-Reference: OS/400 Licensed Program x Function	
OS/400 Licensed Program	**Function**
TCP/IP Connectivity Utilities (5763-TC1)	Pascal TCP and UDP API
	Telnet
	SMTP
	FTP
	LPR/LPD
	POP3 **1**
	HTTP server **1**
	WSG **1**
	DB2WWW **1**
TCP/IP File Server Support/400 (5798-TAA and 5798-TAZ)	NFS (Network File Server)
AS-IS Service Offering	Gopher

1 New in V3R2

2.4.2 Basic Hardware Requirements

The hardware requirements have a direct relationship with two points:

- What type of connection you are going to use (SLIP, PPP, X.25, and so on)?
- How does your ISP provide that type of connection?

Table 2. Cross-Reference: Hardware x Connection Type		
Type of Connection	**Hardware Needed**	**External Hardware Needed**
X.25	Communications port	None
Frame Relay	Communications port	None
SDDI	SDDI IOP	None
FDDI	FDDI IOP	None
Wireless	Wireless IOP	None
SLIP	Communications port	None
PPP	LAN Adapter (Token-Ring, Ethernet, Wireless)	IP Router

2.5 Applying to InterNIC for a Domain and IP Address

To connect to the Internet, you need an IP address; the entity that manages these addresses is the InterNIC. Most ISPs contact the InterNIC for you as part of the service you are buying.

In the past, Internet IP address allocation was performed by the registration services function of the InterNIC. However, as Internet usage increased, the InterNIC delegated responsibility for IP address allocation to the individual Internet Service Providers (ISP). Today the InterNIC no longer allocates IP addresses directly to individuals. Instead it works with domain administrators,

network coordinators, and internet service providers to register Internet domain names and networks.

Organizations that want to obtain an Internet address are encouraged to work through an ISP. The ISP provides one or more valid Internet IP addresses plus other configuration information needed to connect to the Internet. In addition, the ISP can probably also assist with network planning, administration, and security.

The InterNIC is a project sponsored by NSF (National Science Foundation) to provide services to the Internet community. Three organizations collaborated on this project; General Atomics/CERFnet provides information services, AT&T provides directory and database services, and Network Solutions, Inc. (NSI) provides registration services.

The InterNIC Registration Services is located at Network Solutions, Inc., Herndon, VA, and provides assistance in registering networks, domains, ASNs, and other entities to the Internet community through telephone, electronic mail, and U.S. postal mail.

Registration Services works closely with domain administrators, network coordinators, Internet service providers, and other various users to register Internet domains, ASNs, and networks.

Databases and information servers of interest to network users are provided, including the WHOIS registry of domains, networks, and ASNs. Gopher and Wais interfaces are also available for retrieving information and accessing WHOIS. Online documents maintained at registration services include registration related RFCs, registration templates.

Registration Services registers domains, assigns IP network numbers and Autonomous System Numbers (ASNs), and produces the domain zone files for the community. Registration Services also provides assistance to user′s concerning policy and the status of their existing registration request.

Registration Services is responsible for generating and installing the DNS files into the NS.INTERNIC.NET root server.

The InterNIC registration services are necessary to maintain a unique IP addressing worldwide.

For more information, you can access the InterNIC Home Page:
http://www.internic.net/

IBM offers ISP service as part of our Internet Connection family of service offerings. For more information within the U.S., call 1-800-IBM-4YOU (1-800-426-4968) or your local IBM office.

There is one final consideration about IP addresses and the Internet that we should discuss. A block of Internet network addresses has been allocated by the Internet network authority for use only in private networks. IP addresses in this range are guaranteed to never be used as a valid host IP address on the Internet. If an IP router on the Internet sees data for a system in this address range, it is supposed to silently reject (filter) it.

You may want to use IP addresses in this range for some or all of the addresses in your intranet. Doing this should prevent unauthorized access to those portions of your network by people without authorization (hackers) on the Internet.

The address ranges that are reserved for private intranets are:

10.0.0.0 - 10.255.255.255
172.16.0.0 - 172.31.255.255
192.168.0.0 - 192.168.255.255

For more information, see Request For Comment (RFC) 1597, ″Address Allocation for Private Internets″ http://ds.internic.net/rfc/rfc1597.txt

2.6 TCP/IP Domain Name Server

Although the hosts are known by their IP addresses, humans tend to be more comfortable when using the name of a host. In the TCP/IP world, the Domain Name System (DNS) is a distributed database system that provides the mapping between IP addresses and host names. The term *distributed* is used because no single site on the Internet knows all of the information. Each site maintains its own database and runs a database or name server that other systems across the Internet can access.

The AS/400 system cannot be a DNS Server, but can make use of a DNS in the Network. That is, the AS/400 system can be a DNS client. See Figure 6

```
                          Change Remote Name Server
                                                              System:   SYSNM001
Type choices, press Enter.

  Server address  . . . .   9.5.100.76          Internet address
                            9.5.100.75
                            ____________
  Server port . . . . . .   53                  1-65535
  Server protocol . . . .   *UDP                *UDP, *TCP
  Retries . . . . . . . .   3                   1-99
  Retry interval  . . . .   2                   1-99 (seconds)
  Searched first  . . . .   *REMOTE            *REMOTE, *LOCAL

                                                                          Bottom
F3=Exit   F12=Cancel
```

Figure 6. Configuration of a DNS for AS/400 System

Sometimes it is more useful to have your own name server. The local name server is useful, for example, for:

- Private TCP/IP Networks with more than one site and connected through a low speed link.
- Also, if you have your TCP/IP network connected to the Internet through a low speed link, sometimes it can take a long time to get the IP address for a host name.

Chapter 3. Connecting to Your ISP

As described in the previous chapter, there are many different connectivity options for the AS/400 system to link to an ISP. This chapter focuses on how to implement LAN-to-ISP through IP Routing (see 3.1, “LAN-to-ISP”) and Async Dial-Up (see 3.2.1.3, “AS/400 SLIP Support” on page 29).

3.1 LAN-to-ISP

If your AS/400 system already has a LAN interface, you probably want to use the LAN connection to link to the Internet rather than a single-host connection. With a single-host connection, you can connect your AS/400 system directly to the WAN interface using its built-in communications line. With the LAN connection, however, you need an IP router. It has a LAN and the WAN interface that reroutes the IP packets coming from your local subnet to the serial or WAN interface finally to the ISP′s IP router. For a more detailed description of the role of an IP router, refer to *Accessing the Internet*, SG24-2597. Also, when connecting your AS/400 LAN to the Internet, please review the security chapter in this book for more details.

The following illustration might be a typical configuration in which an AS/400 system with a LAN might be connected to the Internet. The AS/400 system on a token-ring LAN connects to the ISP through an IP router and a T1 connection.

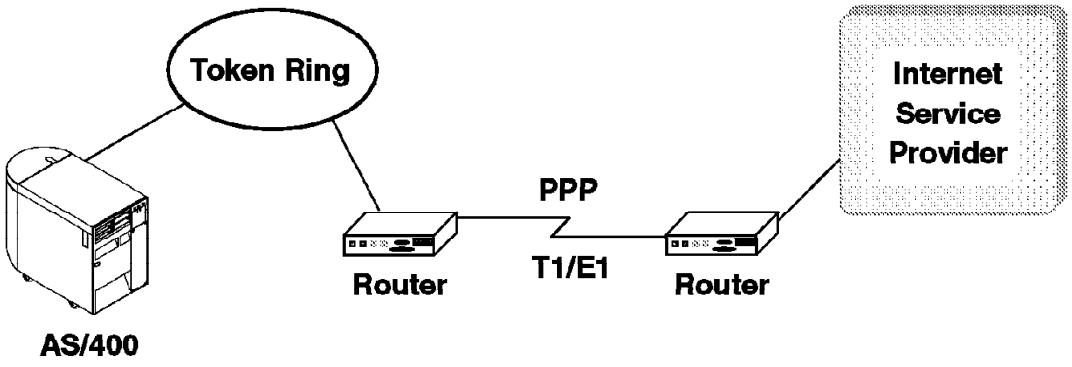


Figure 7. AS/400 System Connecting to an ISP through LAN-to-LAN

In the next few sections, we explore the basic steps to set up this environment.

3.1.1 Setup of IP Router

There are many different routers to chose from; some popular ones are IBM 2210, 3COM, and Wellfleet. Some ISPs provide a complete service including the supply and installation as well as configuration of the router. The following are the steps required if you decide to ″do-it-yourself″:

- After deciding on a router, the next question is speed. Most routers are configured to connect at speeds of 56 Kbps or T1. The speed should be determined by the amount of usage expected.

- The LAN port of your router needs its own IP address, which is supplied by the ISP. This address determines the subnet of your local LAN. The AS/400 system has to have an address in that subnet address range. We explore the configuration needed for the IBM 2210 Router.

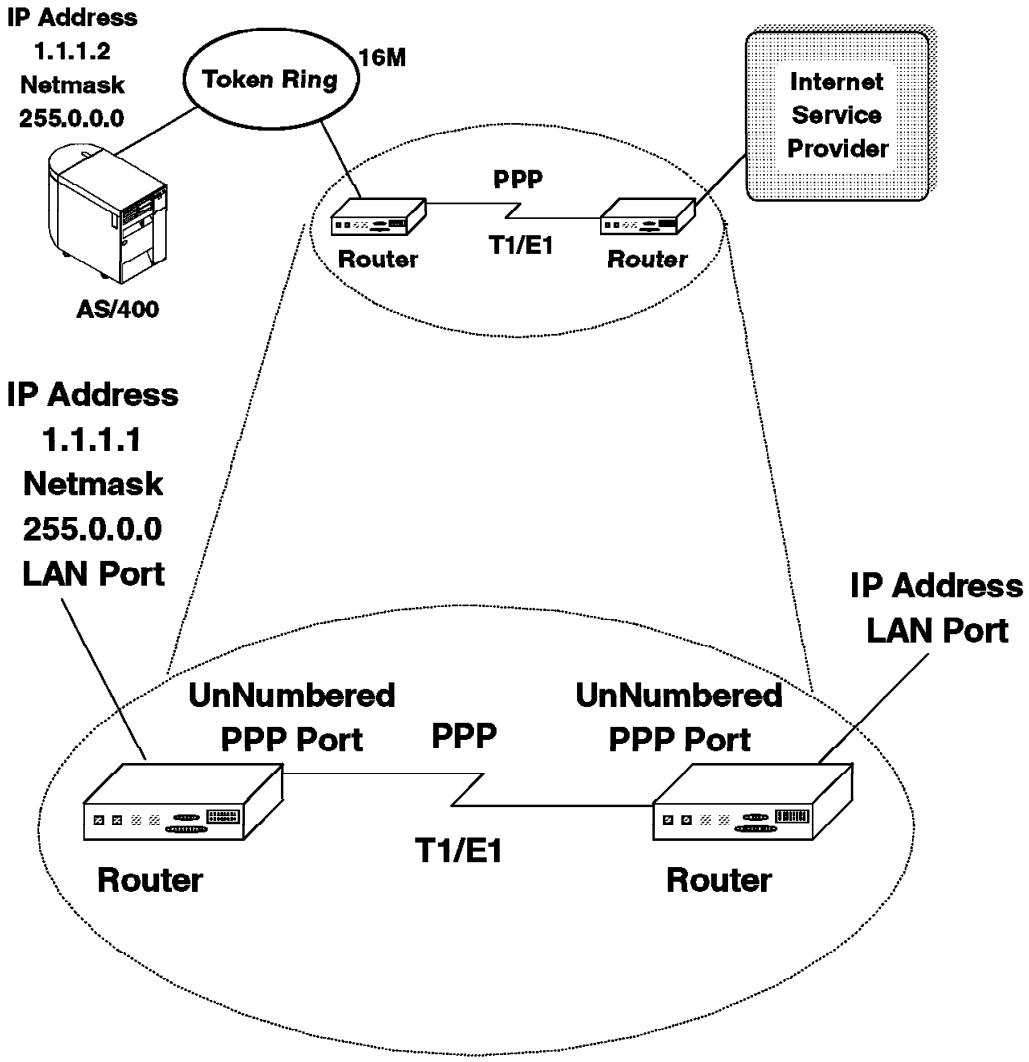


Figure 8. An Expanded View of Our Router Configuration

3.1.1.1 Configuring an IBM 2210 Router

In our sample network configuration, we use IBM 2210 routers. The IBM 2210 router comes with the Nways Multiprotocol Routing Network Services (MRNS) software. It has the following three components:

- The code that provides the routing, bridging, data link switching, and SNMP agent functions.
- The configuration program that offers a graphical user interface that allows you to configure the IBM 2210 from a workstation.
- A monitoring system that allows you to perform network management, problem determination, and configuration.

With the IBM 2210 router, you can support many LAN and WAN interface types such as, (this is not the complete list):

- 10Mb Ethernet LAN
- 4 or 16Mb Token-Ring LAN
- Serial port speeds of 2400 bps to 2.048 Mbps
- Serial ports for V.35/V.36, X.21, and EIA232-D/V.24 electrical specifications
- Frame relay, Point-to-Point Protocol (PPP), and X.25 WAN transports

The IBM 2210 router can be pre-configured at the factory for your particular network requirements, thus saving you the configuration steps.

In our sample network, we only concentrate on how to configure a PPP interface.

3.1.1.2 2210 Configuration

The 2210 has two different methods to configure the router. One method is an ASCII interface in which a ASCII terminal or terminal emulator is cabled to the service port on the rear of the 2210, the other is a Windows based configurator. The Windows based configurator must be completed on a Windows based machine and then sent to the 2210 through the Trivial File Transfer Protocol (TFTP).

For our small network, we used the ASCII interface. In order to connect a PC or ASCII terminal, you can use the special cables shipped as a feature code of the IBM 2210. The IBM 2210 can be configured quickly with the QCONFIG function of the ASCII interface. The QCONFIG function is entered automatically when you first bring up the router since there is no configuration present. Once you are in the Quick Config process, here is what you can configure. The configuration options selected for our sample network are in **bold**.

1. Devices (interfaces)
   - Token-Ring
     - Speed of token-ring (**16**, 4)
     - Connector type (**STP**, UTP)
   - Ethernet (Only appears if you are using an Ethernet model)
     - Connector type (AUI, 10BaseT)
   - Serial
     - Encapsulation (**PPP** or Frame Relay)
     - Cable type (RS232, V.24, **V.35**)
2. Bridging (appropriate types of bridging for your model router). In our scenario, bridging is not necessary.
3. IP
   - Addresses and masks for all interfaces:
     - Address 1.1.1.1 with a network mask of 255.0.0.0 for interface 0 that is the token-ring.
     - Address 0.0.0.1, this sets an unnumbered address for the first serial port of the router. This unnumbered interface scheme saves TCP/IP addresses by not having to subnet your network onto the serial interface.
   - **RIP** or OSPF

- SNMP is automatically configured.

4. IPX (in our example, we are not using the IPX protocol so you may answer NO to this question.)
   - For LAN interface:
     - Ethernet or token-ring encapsulation
     - Network number
   - For WAN interface:
     - Network number
5. Booting:

   Gives previous boot info and asks if you want to create an IBD boot record using this information. **Answer NO to this question**.
6. Asks if you want to *Enable Console Modem Control.* **Answer NO to this question**.
7. Asks if you want to restart the router. Answer YES if you want the configuration you just entered to take effect immediately. If you need to do more configuring, answer NO.

For more details on the commands and setup of the IBM 2210, see: *The Nways MRNS Software's User's Guide*, SC30-3681, or *Protocol Configuration and Monitoring Reference*, SC30-3680.

3.1.1.3 Configuring TCP/IP on the AS/400 System

The following are the configuration details on the AS/400 system:

1. Specify the Internet address of your AS/400 system associated with the LAN interface. To access the following display, you can enter CFGTCP on the AS/400 command line and then select option 1.

```
                          Work with TCP/IP Interfaces
                                                              System:   AS400
Type options, press Enter.
  5=Display   9=Start   10=End

     Internet          Subnet             Line      Line
Opt  Address           Mask               Description  Type

__   _____________
__   127.0.0.1         255.0.0.0          *LOOPBACK    *NONE
__     1.1.1.2         255.0.0.0          TRNLINE      *TRLAN

                                                                        Bottom
F3=Exit      F5=Refresh   F6=Print list   F11=Display interface status
F12=Cancel   F17=Top      F18=Bottom
```

Figure 9. Work with TCP/IP Interfaces Display

2. Specify the IP address of the router as the default routing entry on the AS/400 system (next hop). This tells the AS/400 system to look for this router if it cannot find a TCP/IP address on its own local network.

```
                          Work with TCP/IP Routes
                                                              System:   AS400
Type options, press Enter.
  5=Display

                                                 Type
       Route                 Subnet              of                Next
Opt    Destination           Mask                Service           Hop
 _     ________________      ________________    ___________       _

 _     *DFTROUTE             *NONE               *NORMAL             1.1.1.1
                                                                          Bottom
F3=Exit   F5=Refresh   F6=Print list   F12=Cancel   F17=Top   F18=Bottom
```

Figure 10. Default Route Definition on the AS/400 System

3. Specify a remote name server. The ISP should be able to provide you with an address of a remote name server so that when you try to access a host on the Internet, the remote name server resolves the name to an Internet TCP/IP address. The remote name server can be specified by entering CFGTCP on the AS/400 command line and selecting option 13. The following is an example of the name server display.

```
                        Change Remote Name Server
                                                              System:   AS400
Type choices, press Enter.

  Server address  . . . .   9.5.100.76        Internet address
                            9.5.100.75
                            ________________
  Server port . . . . . .   53                1-65535
  Server protocol . . . .   *UDP              *UDP, *TCP
  Retries . . . . . . . .   3                 1-99
  Retry interval  . . . .   2                 1-99 (seconds)
  Searched first  . . . .   *REMOTE           *REMOTE, *LOCAL

                                                                      Bottom
F3=Exit   F12=Cancel
```

Figure 11. Name Server Definition

4. Specify a host and domain name. When applying for your Internet address, you can also specify a domain name for your network to be known as. This is important if you later want to set up a mail server, for example. Your domain name might be something such as: as400name.yourcompany.com, where your hostname is as400name and your domain name is yourcompany.com. Your host and domain names may be specified by entering CFGTCP on the AS/400 command line and then selecting option 12.

   The following is an example of the Host/Domain name AS/400 display.

```
                    Change Local Domain and Host Names
                                                          System:   AS400
Type choices, press Enter.

  Local domain name . . .   YOURCOMPANY.COM
                                                                        Bottom
F3=Exit   F12=Cancel
```

Figure 12. Host and Domain Name Definition

Refer to the *TCP/IP Configuration and Reference V3*, SC41-3420, for further details.

3.2 SLIP Dial-Up Support on the AS/400

This section deals with the new physical protocol on the AS/400 called SLIP - or Serial Line Interface Protocol. The first section deals with an introduction to the SLIP protocol. The AS/400 supports both dialing out to remote hosts and having remote hosts dial in to the AS/400 and this is described in the next two sections.

Finally, we provide four example scenarios of using the AS/400 SLIP support:

- Please see 3.2.4, “Dial-Out Example to IBM Global Network (IGN)” on page 32.
- Please see 3.2.5, “Dial-Out Example to ISP” on page 38.
- Please see 3.2.6, “Dial-In Example to AS/400 System Using Windows 95 as the Client” on page 39.
- Please see 3.2.7, “Dial-In Example to AS/400 System Using OS/2 WARP as the Client” on page 51.

3.2.1 What is SLIP?

SLIP (Serial Line Interface Protocol) was developed in 1984 by Rick Adams for Berkley UNIX Version 4.2. It became a defacto standard for running TCP/IP over point-to-point serial connections. With the advent of high-speed dial-up modems, the interest in serial line communications grew dramatically, for example, for connecting home computers to the Internet through the RS232 serial port. Thus, it became necessary to develop a standard physical layer protocol for serial lines.

SLIP is a very simple protocol that merely defines a framing method for IP packets flowing over a serial line. It is described in RFC 1055 as a *Nonstandard for Transmission of IP Datagrams Over Serial Lines* , which suggests that it is not an Internet standard. SLIP has some major deficiencies in the following areas that are also described in the RFC.

- Addressing: Both computers in a SLIP link need to know each other’s IP addresses for routing purposes. SLIP currently provides no mechanism for hosts to communicate addressing information over a SLIP connection.

- Packet type identification: Thus, only one protocol can be run over a SLIP connection. While SLIP is a ″Serial Line IP″, if a serial line connects two computers, those computers should be able to use more than one protocol over the line if it is needed. SLIP does not allow them to do so.

  To contrast SLIP′s inability to allow more than one protocol over the single serial line (for example, a LAN protocol such as token-ring), token-ring allows many protocols such as SNA, TCP/IP, IPX, and so on, to all physically share one adapter and coexist down at the physical wire. SLIP only allows IP packets to be carried across the serial line.

  Another protocol that allows IP traffic over serial lines is point-to-point (PPP). As you read later in section 3.3, “Point-To-Point (PPP)” on page 55, PPP allows more than one protocol to share the same serial line.

- Error detection/correction: Noisy telephone lines corrupt packets in transit. Because the line speed is relatively slow retransmitting, a packet is very expensive. Error detection is not absolutely necessary at the SLIP level because any IP application should detect damaged packets (IP header and UDP and TCP checksums should be sufficient). Because it takes so long to retransmit a packet that was corrupted by line noise, it is efficient if SLIP can provide some sort of simple error correction mechanism of its own.

- Compression: Because dial-in lines are relatively slow, packet compression results in large improvements in packet throughput.

3.2.1.1 The Future of SLIP

Despite its deficiencies, SLIP is usually considered adequate. In cases where you only need to run one protocol over the serial line and where the addresses of both computers on the SLIP link are known, then SLIP is adequate. Also, most modems today are capable of doing their own error detection and compression. The IBM Global Network uses SLIP as its serial connection standard and on most UNIX systems, SLIP is part of the operating system.

The point-to-point protocol (PPP) that was developed after SLIP is an official Internet protocol and corrects the deficiencies in SLIP. It is expected to replace SLIP in the near future. For a look at PPP, please see 3.3, “Point-To-Point (PPP)” on page 55.

3.2.1.2 Why Use SLIP?

SLIP is so popular because it allows dial-up access to the Internet through an asynchronous RS232 port, the most common interface that ISPs support. A SLIP connection can be considerably less expensive than a direct Internet connection. A customer might want to use SLIP instead of another interface method if cost is the primary factor. Depending on your location, your ISP may have a local telephone access number. Of course, SLIP connections are much slower than directly-connected AS/400 systems, but if you only need occasional access or if speed is not a factor, then a SLIP connection can be a good choice.

3.2.1.3 AS/400 SLIP Support

The following list provides some planning information for SLIP support.

- Limitations or features:
  - SLIP is available for both dial-in and dial-out support.
  - A default dial-out SLIP script is provided for an Advantis connection:

- This default setting is because of different connection scripts required for each ISP. Customers may work with their individual ISPs to customize additional connection scripts.

– VJ compression for header information on slow async links.

  VJ TCP Header Compression is defined in Internet RFC 1144. Note that VJ header compression does just that; compresses the IP header of the datagram being sent between the two hosts connected by SLIP. It does not compress the data. VJ header compression is also know as Compressed SLIP or CSLIP.

  Most asynchronous modems that you can purchase today have some form of data compression built in, making the need to compress the data as part of the protocol that rides on top of the modem less likely and even unnecessary.

– Use of AS/400 configuration displays.

• Hardware requirements:

  – The SLIP connection is can currently use the following adapter types.

    - 2609 - Two-line EIA 232/V.24 adapter
    - 2612 - One-line EIA 232/V.24 adapter
    - 6152 - Two-line EIA 232/V.24 adapter

    The current set of IOPs that SLIP can run on are the 2623 and the MFIOPs from the D02, D25, all E models, all F models, and all Advanced Series models. There are some unsupported combinations. For details, see the *AS/400 Advanced Series Handbook*.

    Use the Work with Communication Resources (WRKHDWRSC TYPE(*CMN)) command to determine what communication adapters you currently have on your AS/400 system.

  – A variety of well-known (Hayes compatible) ASCII modems is supported.

    Using the current stage, Async IOP/IOA speeds of 1200, 2400, 4800, 9600, 14400, and 19.2 Kbps are supported. The ISP support of 19.2 Kbps is rare so the most common highest denominator is 14400 bps. 28.8 Kbps modems can be used, but the effective throughput is gated by the maximum DTE speed of 19.2 Kbps. The AS/400 system uses hardware flow control to ensure that data over-runs and under-runs do not occur due to error re-transmission and data compression down at the modem level.

3.2.1.4 SLIP Scripts and Standardization

The implementation of SLIP is not standardized around the world. This presents the need for customized SLIP scripts. SLIP scripts are used to pass parameters such as USERIDs and passwords, among others, to your host system. In the same way that your OS/2 Warp PC needs a SLIP script to connect to Advantis or other ISP, the AS/400 system needs that as well.

Note: All the examples of SLIP connections in this chapter can be found in source physical file members in library ITSOIC400 file SLIPSCRIPT.

3.2.2 AS/400 System as a SLIP Client

The AS/400 system is able to dial out to an Internet Service Provider (ISP) using SLIP. The following picture shows how this is accomplished:

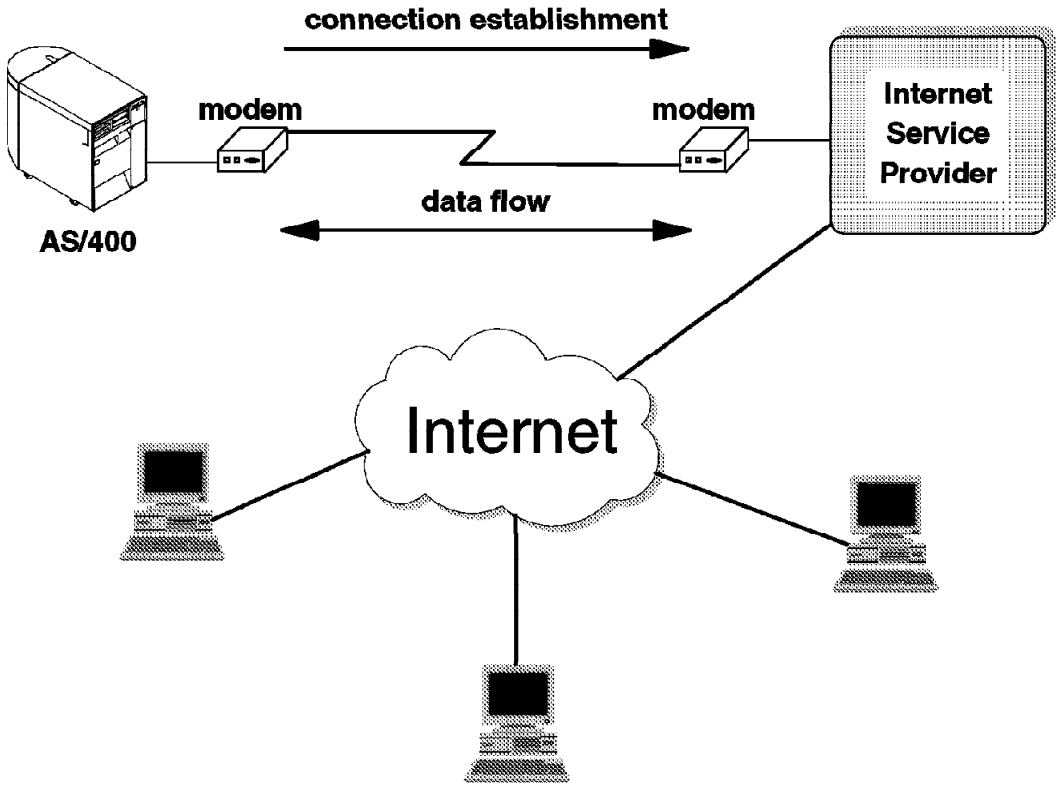


Figure 13. AS/400 System Connected Through Internet Service Provider (ISP)

Note: The AS/400 system can use SLIP to dial into the Internet through an ISP. This provides Internet connectivity for the local AS/400 users. But, the standard connection process to an ISP involves the dynamic assignment of an IP address by the ISP to the dial-up client. This address has no relation to the IP addresses of the remote clients attached to the AS/400 system. Thus, there is no way for any machine on the Internet to know how to route data back to these clients. From the Internet's perspective, the only machine accessible through the SLIP line is the single AS/400 system.

One exception to this is if your ISP permanently assigns a range of IP addresses to you. You can then assign these addresses to your AS/400 system and the remote clients. And since the addresses were assigned by the ISP, they should be known to the rest of the Internet.

Please see 3.2.4, "Dial-Out Example to IBM Global Network (IGN)" on page 32 and 3.2.5, "Dial-Out Example to ISP" on page 38 for examples of SLIP dial-out.

3.2.3 AS/400 System as a SLIP Server

The AS/400 system is also able to accept dial-in connections through SLIP. The following picture shows how that is accomplished.

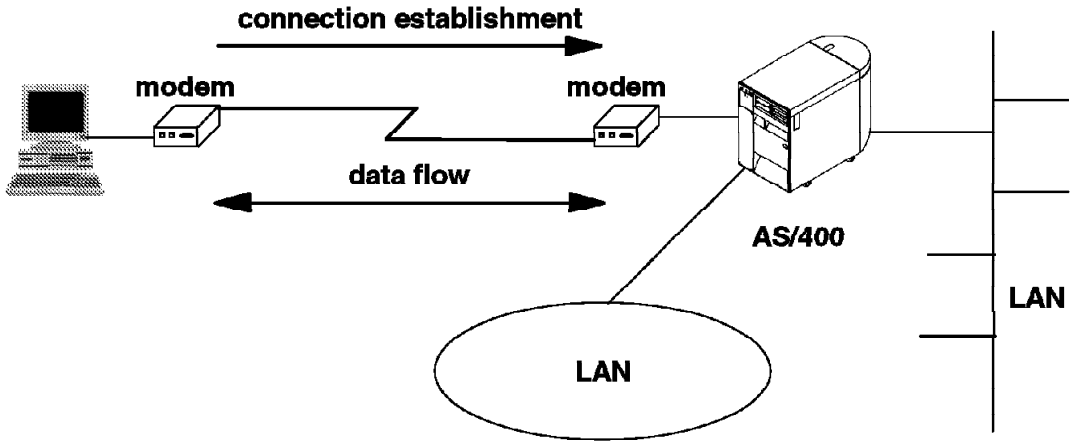


Figure 14. AS/400 System as a SLIP Server

The AS/400 system is able to accept dial-in connections and loosely emulate the role of a ISP. Other systems capable of SLIP dial-out are able to dial into the AS/400 system. The AS/400 system is then capable of routing the dialed-in client anywhere on the AS/400 systems-attached TCP/IP network, including out to the Internet if the AS/400 system was directly attached.

Please see 3.2.6, “Dial-In Example to AS/400 System Using Windows 95 as the Client” on page 39 and 3.2.7, “Dial-In Example to AS/400 System Using OS/2 WARP as the Client” on page 51 for examples of SLIP dial-out.

3.2.4 Dial-Out Example to IBM Global Network (IGN)

This section will guide you through the configuration needed to connect your AS/400 to the IBM Global Network (IGN). It first covers the AS/400 line description configuration and then leads you through the configuration steps for an outgoing SLIP connection to IGN.

3.2.4.1 AS/400 Line Description Configuration

Figure 15 on page 33 shows the parameters that should be used for an asynchronous line that is attached to a modem.

```
CRTLINASC  LIND(ASCSWIT6)
           RSRCNAME(LIN111)
           CNN(*SWTPP)
           LINESPEED(19200)
           SWTCNN(*DIAL)
           AUTOANS(*NO)
           AUTODIAL(*YES)
           DIALCMD(*OTHER)
           INACTTMR(*NOMAX)
           MAXBUFFER(1500)
```

Figure 15. Line Description for Dial-out

ASCSWIT6 was our line description name and LIN111 was our resource name.

3.2.4.2 AS/400 SLIP Configuration

To start the AS/400 SLIP configuration, type the WRKTCPPTP command. The following display is shown:

```
                      Work with Point-to-Point TCP/IP

 Type option, press Enter.
   1=Add    2=Change  3=Copy  4=Remove         5=Display details  6=Print
   9=Start  10=End    12=Work with line status  14=Work with session job

                                                 Line    Line  Job
OptName      Mode Type Status        DescriptionType  Name
 1   IGN        *DIAL
    CAROLANS1 *ANS *SLIPINACTIVE      ASCSWIT7   *ASYNCQTPPANS545
    CAROLDIAL *ANS *SLIPINACTIVE      ASCSWIT8   *ASYNC
    FRANK     *ANS *SLIPINACTIVE      ASCSWIT8   *ASYNC
    FRANKPA   *ANS *SLIPINACTIVE      ASCSWIT8   *ASYNCQTPPANS509
    FRANKPA5  *ANS *SLIPINACTIVE      ASCSWIT5   *ASYNCQTPPANS480
    FRANKPA7  *ANS *SLIPINACTIVE      ASCSWIT7   *ASYNC
    FRANKPP   *ANS *SLIPINACTIVE      ASCSWIT6   *ASYNC
    FRANK5    *ANS *SLIPINACTIVE      ASCSWIT5   *ASYNC
    FRANK7    *ANS *SLIPINACTIVE      ASCSWIT7   *ASYNC
    FRED      *ANS *SLIPRINGW         ASCSWIT5   *ASYNCQTPPANS597
    GARYPA    *ANS *SLIPINACTIVE      ASCSWIT6   *ASYNC
                                                                  More...
 F8=Work with modems  F9=Command line  F10=Local interface status
 F11=Display text     F12=Cancel  F14=Work with session jobs  F24=More keys
```

Figure 16. Work with Point-to-Point TCP/IP

Select **1** for Add, type in a new name (in our case, it was *IGN*), then type *DIAL. Press Enter.

The following figure shows the display after we filled in the parameters for our environment.

```
                Add TCP/IP Point-to-Point *DIAL Profile
                                                        System: AS008
Name: IGN
Text

 Type choices, press Enter.

TCP/IP information:
  Protocol type . . . . . . . . . . : *SLIP
  Local interface address . . . . . .  *DYNAMIC        Address, *DYNAMIC
  Remote IP address . . . . . . . . .  *DYNAMIC        Address, *DYNAMIC
  Request header compression  . . . .  Y               Y=Yes, N=No
  Maximum transmission unit . . . . .  1006            576-1006
  Add default route . . . . . . . . .  Y               Y=Yes, N=No
  Additional name server  . . . . . .  165.87.194.244  Address, *NONE

                                                                    More...
 F2=Change modem information  F3=Exit  F4=List  F9=Command line
 F12=Cancel
```

Figure 17. TCP/IP Information

1. Local interface address **(Required)**:
   - *DYNAMIC: This means that the local IP address is determined by the server system we are connecting to and is passed back by the server as part of the connection dialog.
   - This value can also be statically configured if the remote system is *not* going to pass back the address to the AS/400 system.
2. Remote IP address **(Required)**:
   - *DYNAMIC: This means that the remote IP address is determined by the server system we are connecting to and is passed back by the server as part of the connection dialog. This is the value that is used as the ″NEXT HOP″ value for any defined routes to the remote system.
   - This value can also be statically configured if the remote system is *not* going to pass back the address to the AS/400 system.
3. Request header compression: The default is ″Y″. This is to allow better performance over slow serial lines and is set by the Client (DIAL) side.

   Note: We suggest that you use IP header compression in almost all cases. In fact, some Internet Service Providers (ISPs) require the use of IP header compression even though the RFCs clearly says that the client gets to choose whether or not IP header compression is performed.
4. Maximum transmission unit:
   - 1006 is the maximum MTU (Maximum Transmission Unit) value allowed for SLIP. The SLIP MTU cannot be larger than this. The exact default value to be used is determined by the results of testing with the probable typical customer configuration.

     Note: The **MAXBUFFER** parameter for the CRTLINASC command can be used to specify the maximum size of the line′s inbound and outbound data buffers. The default for the MAXBUFFER parameter is 896 bytes. The value specified for the SLIP MTU is required to be less than or equal to the value that was specified for MAXBUFFER.
5. Add default route:

- Specifying ″Y″ for this value instructs TCP/IP to automatically add a default route to the IP address provided as the ″Remote IP address″ of this connection.
- A session that has this option set to ″Y″ is only allowed to start if no default routes are defined. If a default route does exist, the start fails and an escape message is issued.
- The default value for this option is ″N″.
- Only **one** dial session may have a default route defined. If another dial session is started with this field set to ″Y″, then it is not allowed to start.

6. Additional nameserver:

   - This field can be used to specify the IP address of a name server of the remote domain. It provides limited support when connecting to a remote network. If a name server address is provided, it is automatically added at the end of any existing list of name servers as shown in CFGTCP menu option 13.
   - The default value for this option is *NONE.

   Important Note

   Most sockets applications that were activated prior to establishing the *DIAL connection to a remote system do *not* see this name server unless they are ended and restarted. In particular, SMTP needs to be restarted after establishing a remote connection for it in order to use a name server that has been dynamically added.

Press the Page Down key to see the second part of the *DIAL profile configuration. Figure 18 shows the display after we filled in the specific parameters for our environment.

```
                    Add TCP/IP Point-to-Point *DIAL Profile
                                                          System:   AS008
Name:   IGN
Text    ____________________________________________

Type choices, press Enter.

Physical line information:
  Line description  . . . . . .   ASCSWIT6      Name
  Line type . . . . . . . . . . :  *ASYNC
  Autocreate controller and device Y             Y=Yes, N=No
    Remote location name  . . . . .  ________    Name

Modem information:
  Use a modem . . . . . . . . . .  Y             Y=Yes, N=No
    Modem information name                       F4 for list
      IBM  9600 7855 ____________________________________________

                                                                  More...
F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 18. Physical Line and Modem Information

Physical Line Information

1. Line description name **(Required)**
   - A valid line description is entered here. Only lines of type *ASYNC are supported.

Modem Information

1. Use a modem:
   - We selected ″Y″ (Yes).
2. Modem information name:
   - We selected the *F4 for list* function key to display a Pop-up selection list of previously-defined modem strings.

```
                 Add TCP/IP Point-to-Point *DIAL Profile
                                                        System: AS008
Name: IGN
Text

 Type choices, press Enter.

Script source information:
  Use connection dialog script  . . .  Y              Y=Yes, N=No
    Member  . . . . . . . . . . . . .  DIALIGN        Name
    File  . . . . . . . . . . . . . .  QATOCPPSCR     Name
      Library . . . . . . . . . . . .    QUSRSYS      Name
    ASCII character set identifier     00819          1-65533, *DFT

                                                                    More...
 F2=Change modem information  F3=Exit  F4=List  F9=Command line
 F12=Cancel
```

Figure 19. ″Use Connection Script″ = ″Y″ Display

Script Source Information

1. Use connection dialog script:
   - We specified ″Y″ for yes.
   - Member/file/library:
     - Use these files to define where to find the connection script used to allow the AS/400 system to dial a remote system or ISP. This script is for the dialer. The remote server system must have a compatible script.
     - DIAL400 is a script used to dial another AS/400 system and is the default.
     - Another valid member is DIALIGN that is used to dial the IBM Global Network.
     - The user can also create their own scripts to access other Internet Service Providers (ISP) or to dial other systems.
   - ASCII character set identifier:
     - The ASCII CCSID is used to translate ASCII to EBCDIC and EBCDIC to ASCII connection script data.

Page down again to see the last configuration display. Figure 20 on page 37 shows the parameters changed to our specific environment.

```
                  Add TCP/IP Point-to-Point *DIAL Profile
                                                          System: AS008
Name: IGN
Text

 Type choices, press Enter.

Remote system access information:
  Remote service phone number
     9,,,,,7661001
  Remote service access name
     u1inet as4edu
  Remote service access password
     xxxxxx

                                                                      Bottom
 F2=Change modem information  F3=Exit  F4=List  F9=Command line
 F12=Cancel
```

Figure 20. Remote System Access Information

Remote System Access Information (Optional)

1. Remote service phone number: The telephone number to call to dial the remote system.
2. Remote service access name: The access name (or user profile for the AS/400 system) that you use to connect to the remote system or ISP.
3. Remote service access password: The password for this access name or user profile.

   Note: The password listed in our example is xxxxxx. All x's are listed to show where the password is entered. The password is not listed here after the profile is added. It is stored in an encrypted form in storage (assuming that system value QRETSVRSEC is set to 1). If QRETSVRSEC is set to 0, it is not possible to create a SLIP profile with a password.

With these new definitions, the AS/400 system is then able to use the low cost SLIP connections many ISPs now provide. The basic steps that occur when attaching through a SLIP connection are as follows:

1. ASCII mode handshaking.
2. Acceptance of ISP's supplied IP address.
3. Transfer of ISP account logon and password information.

The following two examples show an actual dialog between a SLIP client and the IBM Global Network ISP and the conversion into a client connection script for the IBM Global Network:

```
&
*******************************************
Welcome to the IBM Global Network
*******************************************
Enter dial script version ==>
1.1
Gateway: IBMT2YA0 Port: 22
Select one of the following services:
INTERNET
Enter service ==>
INTERNET
Enter account userid password (/new_password) ==>
ulinet barrier password
129.37.3.150 is your IP address.
129.37.1.10 is the Gateway IP address.
Begin TCP/IP communication now.
```

Figure 21. Actual SLIP Dialog Between AS/400 System and IBM Global Network

The connection dialog is scrubbed to eliminate all but the necessary keywords to create a client connection script for the SLIP Session Manager:

```
* IBM Global Network Client Script Example
> version ==>
< (VERSION)
> service ==>
< INTERNET
> password ==>
< ulinet (USERID) (PASSWORD)
> (IPDEST) is your IP address.
> (IPGATE) is the Gateway IP address.
> Begin TCP/IP communication now.
```

Figure 22. Scrubbed Dialog Between IBM Global Network and AS/400 System

3.2.5 Dial-Out Example to ISP

This is very similar to the IGN example. We highlight some of the differences.

3.2.5.1 AS/400 Line Description Configuration

We used the same line description as listed in 3.2.4.1, “AS/400 Line Description Configuration” on page 32.

3.2.5.2 AS/400 SLIP Configuration

We needed to create our own script. Refer to “Creating and Changing Connection Scripts” on page 42 for information on creating and changing scripts.

Further details regarding how to code the scripts is available in the latest version of *TCP/IP Configuration and Reference*, SC41-3420. Figure 23 on page 39 shows our script source information and remote system access information for this profile.

```
               Display TCP/IP Point-to-Point *DIAL Profile
                                                          System: AS008
Name . . . . : ISP
Text . . . . :

Script source information:
  Use connection dialog script  . . . . . . . . . . . : Y
    Member  . . . . . . . . . . . . . . . . . . . . . : ANEXISP
    File  . . . . . . . . . . . . . . . . . . . . . . : SLIPSCRIPT
      Library . . . . . . . . . . . . . . . . . . . . :   HALLEEN
    ASCII character set identifier  . . . . . . . . . : 00819

Remote system access information:
  Remote service phone number:
    9,,,,,,15072828164
  Remote service access name:
    rhalleen
  Remote service password defined . . . . . . . . . . : Y
    Password change date/time . . . . . . . . . . . .   08/23/2812:03:06
                                                                    Bottom
 Press Enter to continue.

 F3=Exit  F12=Cancel
```

Figure 23. Remote System Access Information

Figure 24 shows the script we used to connect to our ISP.

```
*****************************************************
 * CLIENT SCRIPT EXAMPLE FOR CONNECTING TO ISP
 username:
 & (USERID)
 password:
 & (PASSWORD)
 option
 & 3
 Annex address is (IPGATE) Your address is (IPADDR)
*****************************************************
```

Figure 24. AS/400 Client Script for Connection to ISP

3.2.6 Dial-In Example to AS/400 System Using Windows 95 as the Client

This section will guide you through the configuration needed to have a remote Windows 95 client connect to your AS/400. The line descriptions needed for this kind of connection do not change from what you have seen with the dial-out scenarios.

The remaining two sections show you how to first configure the AS/400 side of the connection to accept the incoming call and then show you how to configure the Windows 95 client.

3.2.6.1 AS/400 Line Description Configuration

Please refer to 3.2.4.1, “AS/400 Line Description Configuration” on page 32 for information about this configuration.

3.2.6.2 AS/400 Configuration for Windows 95 Client

To start the AS/400 SLIP configuration, type the WRKTCPPTP command; the following display is shown.

```
                    Work with Point-to-Point TCP/IP

 Type option, press Enter.
   1=Add    2=Change  3=Copy  4=Remove         5=Display details  6=Print
   9=Start  10=End    12=Work with line status  14=Work with session job

                                                Line    Line  Job
OptName      Mode Type Status        DescriptionType  Name
 1   WIN95SCRT  *ANS
    PC        *ANS *SLIPRINGW         MODEMLIN1  *ASYNCQTPPANSO54

 F8=Work with modems  F9=Command line  F10=Local interface status
 F11=Display text     F12=Cancel  F14=Work with session jobs  F24=More keys
```

Figure 25. Work with Point-to-Point TCP/IP

Select **1** for Add, type in a new name (in our case, it was *WIN95SCRT*), and then type *ANS. Press Enter.

The following figure shows the display after we filled in the parameters for our environment.

```
                    Add TCP/IP Point-to-Point *ANS Profile
                                                          System:   AS008
Name:   WIN95SCRT
Text    ____________________________________________

Type choices, press Enter.

TCP/IP information:
  Protocol type . . . . . . . . . . :   *SLIP
  Local interface address . . . . . .   9.5.69.208       Address, F4 for list
  Remote IP address . . . . . . . . .   9.5.69.215       Address
  Maximum transmission unit . . . . .   1006             576-1006
  Allow proxy ARP . . . . . . . . . .   Y                Y=Yes, N=No
  Add default route . . . . . . . . .   N                Y=Yes, N=No

Physical line information:
  Line description  . . . . . . . . .   MODEMLIN2        Name
  Line type . . . . . . . . . . . . :   *ASYNC
  Autocreate controller and device      Y                Y=Yes, N=No
    Remote location name  . . . . . .     ________       Name
                                                                  More...
F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 26. TCP/IP and Physical Line Information

TCP/IP Information

1. Local interface address **(Required)**:

   - The local IP address to use as the gateway address for the remote client. They use this address as a next hop value for a route or default route to the AS/400 system.
   - Use F4 to get a list of already-defined local addresses to use, or enter a new address. If an already-defined local address is chosen, then it can be used for Proxy ARP on behalf of the remote system dialing in. The Proxy ARP flag has to be set to ″Y″ for this to occur.

     Proxy ARP is a technique that allows one machine, the proxy agent, to answer ARP requests that are actually destined for a different machine. Proxy ARP is useful with SLIP because it allows a remote SLIP client to logically appear to be part of a local network (or on the same subnet).

2. Remote IP address **(Required)**:

   - This is the address that the Remote Client should use as *their* local interface address. It is the IP address used to allow the remote system and the local AS/400 system to communicate.
   - If the local IP address chosen already exists and is being used for Proxy ARP, then the remote IP address that you choose must further be defined to be on the *same* subnet as the local IP address that is defined by the subnet mask for the local IP address.

3. Maximum transmission unit:

   - 1006 is the maximum MTU value allowed for SLIP. The SLIP MTU cannot be larger than this. The exact default value to be used is determined by the results of testing with the probable typical customer configuration.

     Note: The **MAXBUFFER** parameter for the CRTLINASC command can be used to specify the maximum size of the line′s inbound and outbound data buffers. The default for the MAXBUFFER parameter is 896 bytes. The value specified for the SLIP MTU must be less than or equal to the value of MAXBUFFER.

4. Allow proxy ARP:

   - The default is ″N″. Set to ″Y″ if proxy ARP is to be used.
   - This field can *only* be set to ″Y″ if the **Local interface address** defined is a true local interface that is already defined and the **Remote IP address** is defined to be on the same subnet as the local address.

Physical Line Information

1. Line description name **(Required)**:

   - A valid line description is entered here. Only lines of type *ASYNC are supported.

2. Autocreate controller and device:

   - If ″Y″ (yes), then TCP/IP creates the appropriate controller and device for the session. When the session is completed, the autocreated controller and device are deleted.

Page down to see the second part of the *ANS profile configuration. The following figure shows the display after we filled in the specific parameters for our environment.

```
                  Add TCP/IP Point-to-Point *ANS Profile
                                                        System: AS008"
Name: WIN95SCRT
Text

 Type choices, press Enter.

Modem information:
  Use a modem . . . . . . . . . . . . .  Y              Y=Yes, N=No
    Modem information name                              F4 for list
       IBM 28800 7852-010

Script source information:
  Use connection dialog script  . . .  Y              Y=Yes, N=No
    Member  . . . . . . . . . . . . .  ANSWIN95A      Name
    File  . . . . . . . . . . . . . .  SLIPSCRIPT     Name
      Library . . . . . . . . . . . .    ITSOIC400    Name
    ASCII character set identifier     00819          1-65533, *DFT


                                                                      More...
 F2=Change modem information  F3=Exit  F4=List  F9=Command line
 F12=Cancel
```

Figure 27. Modem and Source Script Information

Modem Information

1. Use a modem:
   - We selected "Y" (Yes).
2. Modem information name:

   The field "Modem information name" is set to blanks when initially displayed as part of the 1=Add dialog. A valid value must be specified before the profile can be added (when "Use a modem" is "Y").
   - We selected the *F4 for list* function key to display a Pop-up selection list of previously-defined modem strings.

Script Source Information: For this example, we used a script since we wanted an exchange of USERID, password, and IP addresses.

1. Use connection dialog script:
   - We specified "Y" (Yes).
   - Member/file/library:
     - Where to find the connection script to use to allow remote systems to dial into the AS/400 system. This script is for the server. The remote system that dials in must have a compatible script.
   - ASCII character set identifier:
     - The ASCII CCSID is used to translate ASCII to EBCDIC and EBCDIC to ASCII connection script data.

Creating and Changing Connection Scripts: You cannot change the default connection script file QUSRSYS/QATOCPPSCR. You must first create your own connection script file. Do this by copying the default file as follows:

```
CPYF FROMFILE(QUSRSYS/QATOCPPSCR)
     TOFILE(lib/file)
     FROMMBR(*ALL)
     TOMBR(*FROMMBR)
     MBROPT(*ADD)
     CRTFILE(*YES)
```

where lib/file represents your own new file.

Please refer to the latest version of *TCP/IP Configuration and Reference*, SC41-3420 for detailed script information.

The following figure shows the AS/400 script from Member: ANSWIN95A, File: SLIPSCRIPT, and Library: ITSOIC400

```
*************************************************************************
 * SERVER CONNECTION SCRIPT EXAMPLE WITH LOGIN AND PASSWORD / WIN95
 (PROMPT)
 & Userid:
 (USERID)
 & Password?
 (PASSWORD)
 & InternetLR/E>
 (PROMPT)
 & Your address is (IPADDR)
*************************************************************************
```

Figure 28. AS400 Server Script for Our Windows 95 Connection

Page down again to see the third configuration display. The following figure shows the parameters changed to our specific environment.

```
                  Add TCP/IP Point-to-Point *ANS Profile
                                                       System: .SYSNM008
Name: WIN95SCRT
Text

 Type choices, press Enter.

Local system security:
  Allow IP datagram forwarding  . . .  Y               Y=Yes, N=No
  System access authorization list     PCSLIP          *NONE, Name

                                                                    Bottom
 F2=Change modem information  F3=Exit  F4=List  F9=Command line
 F12=Cancel
```

Figure 29. Local System Security

Local System Security

1. Allow IP datagram forwarding:
   - When a remote client connects to the AS/400 system, this value determines whether the TCP/IP stack allows IP datagrams originating from the remote host to be forwarded onto IP addresses *other* than the local IP address defined for the AS/400 system for this connection.

Note: If **IP Datagram Forwarding** is set OFF at the system level through the IPDTGFWD parameter on the CHGTCPA command, then the value for **IP Datagram Forwarding** on each of the SLIPs has *no* effect since IP datagram forwarding is *not* allowed for *any* TCP/IP interface.

2. System access authorization list:

   - *NONE means that when a client is trying to connect to the AS/400 system and they pass a User ID and Password, these values are ignored and the connection is allowed.

     If there is no authorization list, it only means that there is no connection security. There is still application level security such as Telnet USERID and password that is needed.
   - If the user wants to validate that the remote client is allowed to connect, then a valid system authorization list name can be entered here. The authorization list is created using the CRTAUTL command and then adding user profile names that are authorized to connect. The password passed in is also validated to ensure it is the correct password for the user profile.

     We created an authorization list with the name PCSLIP, and added the user profile name HALLEEN to it. The authorization list needs to be created before referring to it here.

     Note: A user is not allowed to set both ″Use connection script″ to ″N″ and ″System access authorization list″ to a value other than *NONE. Providing the name of an (existing) authorization list forces ″Use connection script″ to ″Y″. An information message is issued to explain why this change was made.

Starting the SLIP Profile: The following display shows how to start the SLIP profile.

```
                      Work with Point-to-Point TCP/IP

 Type option, press Enter.
   1=Add    2=Change  3=Copy  4=Remove         5=Display details  6=Print
   9=Start  10=End    12=Work with line status  14=Work with session job

                                                 Line    Line  Job
OptName      Mode Type Status        DescriptionType  Name

    PC        *ANS *SLIPINACTIVE      MODEMLIN1  *ASYNCQTPPANS054
 9  WIN95SCRT *ANS *SLIPINACTIVE      MODEMLIN2  *ASYNCQTPPANS055

                                                                    Bottom
 F8=Work with modems  F9=Command line  F10=Local interface status
 F11=Display text     F12=Cancel  F14=Work with session jobs  F24=More keys
```

Figure 30. Start Session

Select Option 9 to Start the profile. After the profile has successfully started, the status of the profile is *RINGW*. It is necessary to press F5 for Refresh to update the display shown in Figure 30.

If the status stays in STRSSN, there *may* be a message in QSYSOPR.

```
                      Work with Point-to-Point TCP/IP

 Type option, press Enter.
   1=Add    2=Change  3=Copy  4=Remove         5=Display details  6=Print
   9=Start  10=End    12=Work with line status  14=Work with session job

                                                 Line    Line  Job
OptName      Mode Type Status        DescriptionType  Name

    PC        *ANS *SLIPINACTIVE      MODEMLIN1  *ASYNCQTPPANS054
    WIN95SCRT *ANS *SLIPRINGW         MODEMLIN2  *ASYNCQTPPANS055

                                                                      Bottom
 F8=Work with modems  F9=Command line  F10=Local interface status
 F11=Display text     F12=Cancel  F14=Work with session jobs  F24=More keys
```

Figure 31. Waiting for Incoming Call

Figure 31 shows *WIN95SCRT* with a status of *RINGW*.

3.2.6.3 Windows 95 Configuration for SLIP Client

Important Note

Before starting the Window 95 SLIP configuration, verify that TCP/IP is installed, verify that Dial-Up Networking is installed, and verify that Dial-Up Scripting Tool is installed.

We needed to install the Dial-Up Scripting Tool with the following procedure using Windows 95:

- Click on Start button.
- Go to Settings.
- Click on Control Panel.
- Double click on Add/Remove Programs.
- Click on Windows Setup.
- Click on Have Disk.
- Enter [CD drive]:\ADMIN\APPTOOLS\DSCRIPT.
- Click on OK.

The following information lists the configuration on the PC side to make a SLIP connection to the AS/400 system. We are using scripts on both the PC and AS/400 system for greater security.

- On the Windows 95 PC, click the **Start** button.
- Click on **Dial-Up Networking** program in the Accessories folder.
- To get to the initial configuration window, click on **Connections** , then click on **Make New Connection**. The following window is shown:

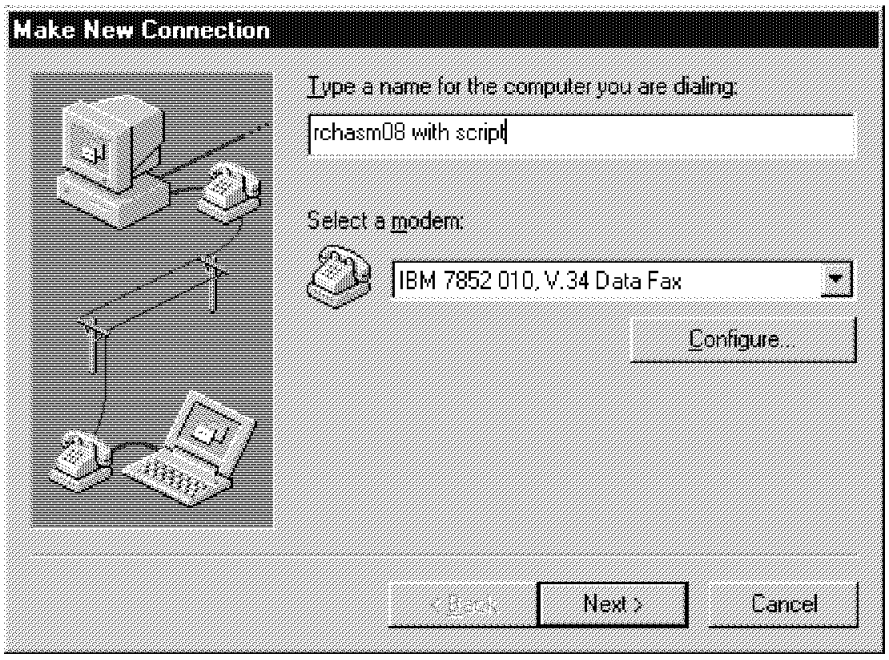


Figure 32. Make New Connection #1

Type a name for the computer you are dialing. The modem was already selected because we had Windows 95 automatically detect our modem when we installed the modem for Windows 95.

- Click on **Next**. The following window is shown:

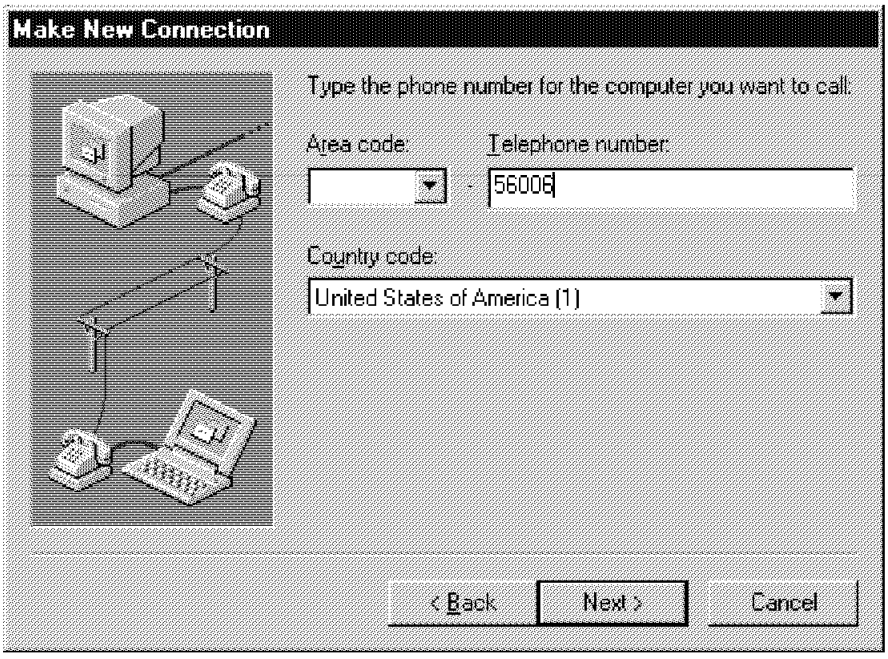


Figure 33. Make New Connection #2

- Fill in the phone number.
- Click on **Next**, which brings up the third Make New Connection window. This window is not shown in this example.

- Click on **Finish**, which saves the configuration in your Dial-Up Networking folder.
- Next, highlight the newly-created connection. Click on **File** and then click on **Properties**. The following window is shown:

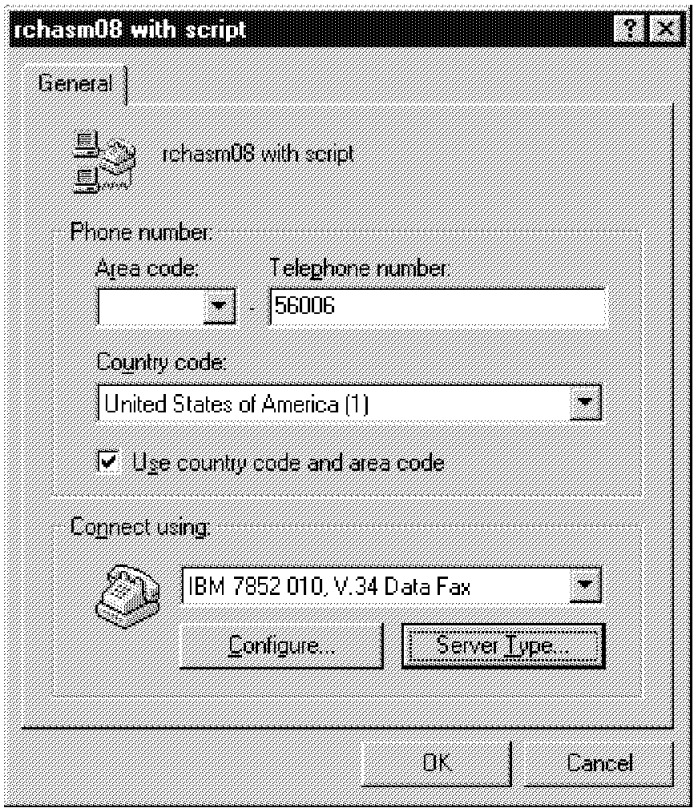


Figure 34. Properties Window

- Click on **Server Type**; the following window is shown:

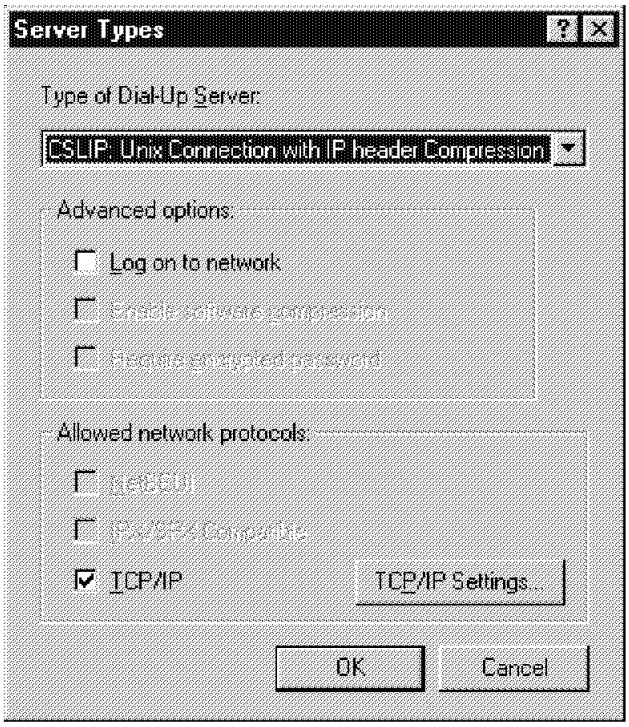


Figure 35. Server Types

On the Server Types menu, do the following:

- Select **CSLIP Unix Connection with IP header Compression**.
- Do *not* check **Log on to network**.

  We used dynamic TCP/IP addresses, so we did *not* need to enter anything for TCP/IP Settings.
- Click on **OK**, then click **OK** again to complete the **General** tab.

The next thing that we did in the process was to assign a SLIP script for the PC. The following steps explain this.

- On the Windows 95 PC, click the **Start** button.
- Click on **Dial-Up Scripting Tool** in the Accessories folder.
- Click on the connection that was just created (in our case, it was **sysnm008 with script**). The following window is shown:

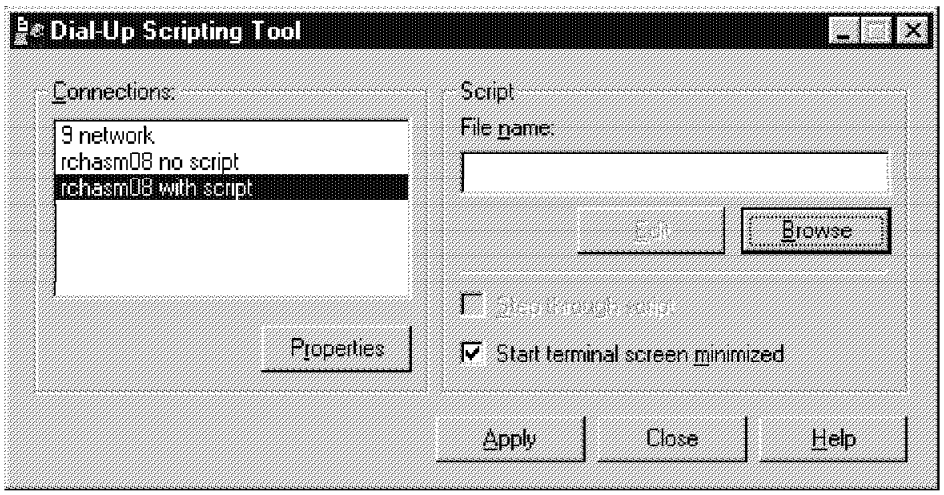


Figure 36. Dial-Up Scripting Tool

- Click on **Browse** to see a list of files.
- Double-click on **Slip** to select file Slip.scp.
- Click on **Apply** to apply the script to this connection.

See Figure 37 on page 50 for the script used on the Windows 95 client. This script has also been included in directory /itsoic.400 with file name slip.scp for your reference. ([M in the script means CTRL-M or Carriage Return.)

```
;
; This is a script file that demonstrates how
; to establish a slip connection with a host.
;
; A script file must have a 'main' procedure.
; All script execution starts with this 'main'
; procedure.
;


; Main entry point to script
;
proc main

   ; Delay for 2 seconds first to make sure the
   ; host doesn't get confused when we send the
   ; two carriage-returns.

   delay 2
   transmit "[M[M"

   ; Wait for the login prompt before entering
   ; the user ID

   waitfor "serid:"
   transmit $USERID
   transmit "[M"

   ; Enter the password

   waitfor "assword?"
   transmit $PASSWORD
   transmit "[M"

   waitfor "InternetLR/E>"
   transmit "slip"
   transmit "[M"

   ; An alternative to the following two lines is
   ;
   ;    set ipaddr getip 2
   ;
   ; since we know that our address is the second one given.

   waitfor "Your address is "
   set ipaddr getip

endproc
```

Figure 37. Windows 95 Default Script File (Slip.scp)

Next, we can make the connection assuming that the AS/400 system is waiting for an incoming call.

- Go back to **Dial-Up Networking**.
- Double-click on Entry name (in our case, it was **sysnm008 with script**).
- Enter the user name and password in the **User name** and **Password** fields.
- Click on **Connect** to make the connection.

Figure 38 on page 51 shows the connection window. Windows 95 shows that we are connected at 28 800, but the AS/400 system is only capable of up to 19 200.

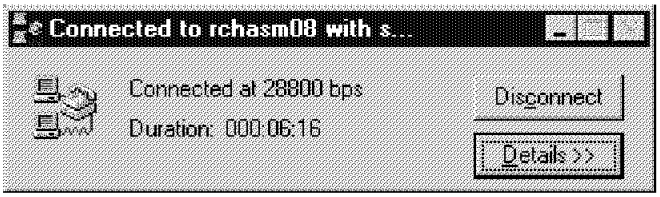


Figure 38. Connected Using SLIP

3.2.7 Dial-In Example to AS/400 System Using OS/2 WARP as the Client

This is an example that does not use scripts. This is similar to the Windows 95 example. We highlight some of the differences.

3.2.7.1 AS/400 Line Description Configuration

Please refer to 3.2.4.1, “AS/400 Line Description Configuration” on page 32 for this configuration.

3.2.7.2 AS/400 Configuration for OS/2 WARP Client:

The following figures show the completed displays that we used for our *ANS profile.

```
                      Work with Point-to-Point TCP/IP

 Type option, press Enter.
   1=Add    2=Change  3=Copy  4=Remove         5=Display details  6=Print
   9=Start  10=End    12=Work with line status  14=Work with session job

                                                 Line    Line  Job
OptName      Mode Type Status        DescriptionType  Name
 1   PC         *ANS

  (No configuration profiles)

                                                                      Bottom
 F8=Work with modems  F9=Command line  F10=Local interface status
 F11=Display text     F12=Cancel  F14=Work with session jobs  F24=More keys
```

Figure 39. Work with Point-to-Point TCP/IP

```
                    Add TCP/IP Point-to-Point *ANS Profile
                                                          System:   SYSNM008
Name:   PC
Text    ____________________________________________

Type choices, press Enter.

TCP/IP information:
  Protocol type . . . . . . . . . . :   *SLIP
  Local interface address . . . . . .   10.5.69.208      Address, F4 for list
  Remote IP address . . . . . . . . .   10.5.69.231      Address
  Maximum transmission unit . . . . .   1006             576-1006
  Allow proxy ARP . . . . . . . . . .   N                Y=Yes, N=No
  Add default route . . . . . . . . .   N                Y=Yes, N=No

Physical line information:
  Line description  . . . . . . . . .   MODEMLIN1        Name
  Line type . . . . . . . . . . . . :   *ASYNC
  Autocreate controller and device      Y                Y=Yes, N=No
    Remote location name  . . . . . .     ________       Name
                                                                    More...
F2=Change modem information   F3=Exit   F4=List   F9=Command line
F12=Cancel
```

Figure 40. TCP/IP and Physical Line Information

```
                  Add TCP/IP Point-to-Point *ANS Profile
                                                      System: .SYSNM008
Name: PC
Text

 Type choices, press Enter.

Modem information:
  Use a modem . . . . . . . . . . . .  Y              Y=Yes, N=No
    Modem information name                             F4 for list


Script source information:
  Use connection dialog script  . . .  N              Y=Yes, N=No
    Member  . . . . . . . . . . . . .  ANS400         Name
    File  . . . . . . . . . . . . . .  QATOCPPSCR     Name
      Library . . . . . . . . . . . .    QUSRSYS      Name
    ASCII character set identifier     00819          1-65533, *DFT


                                                                  More...
 F2=Change modem information  F3=Exit  F4=List  F9=Command line
 F12=Cancel
```

Figure 41. Modem Information

- We chose local and remote IP addresses that were on a new network.
- Allow proxy ARP was not used.
- Connection dialog script was not used.
- Allow IP datagram forwarding was No.

- System access authorization list was *NONE.

3.2.7.3 OS/2 WARP Configuration for SLIP Client

The following figures show how we configured the OS/2 PC.

Double-click on **Network Dialer** in TCP/IP folder. The following window is shown:

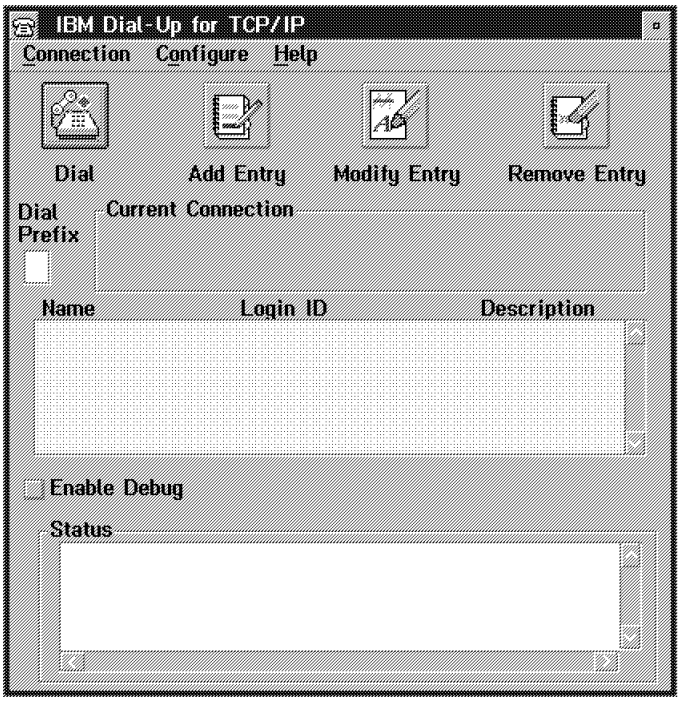


Figure 42. IBM Dial-Up for TCP/IP

Click on **Add Entry**. The following window is shown:

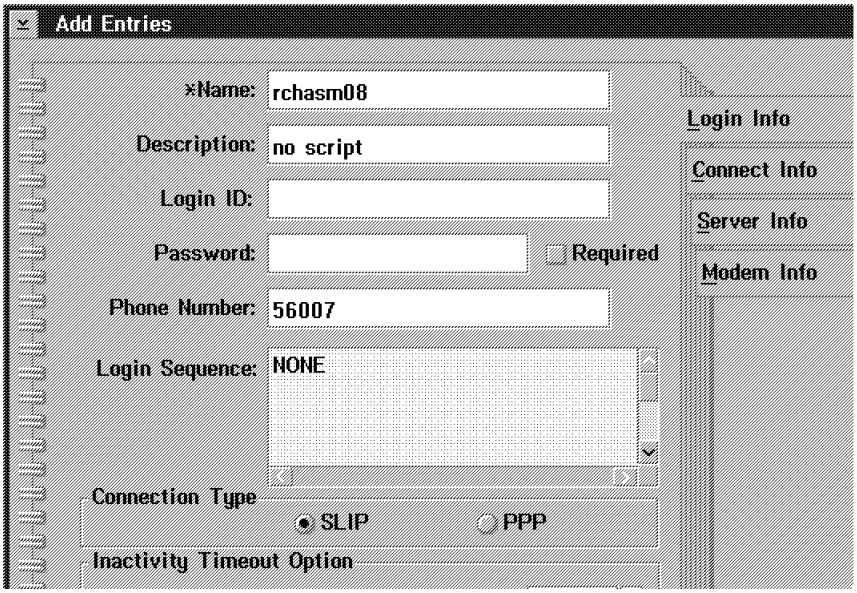


Figure 43. Login Info

Click on **Connect Info**. The following window is shown:

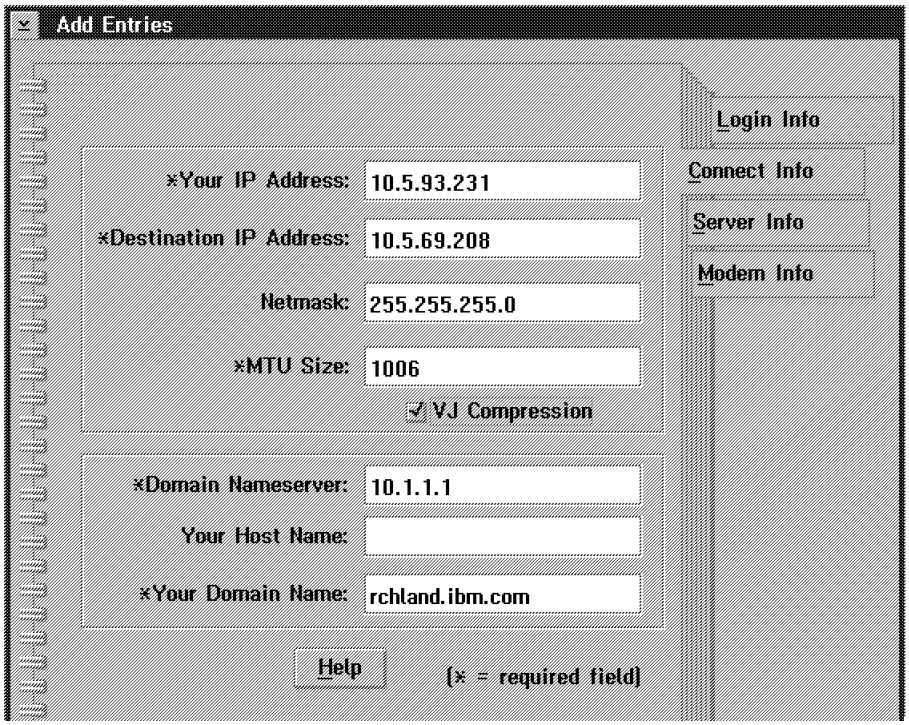


Figure 44. Connect Info

Click on **Modem Info**. The following window is shown:

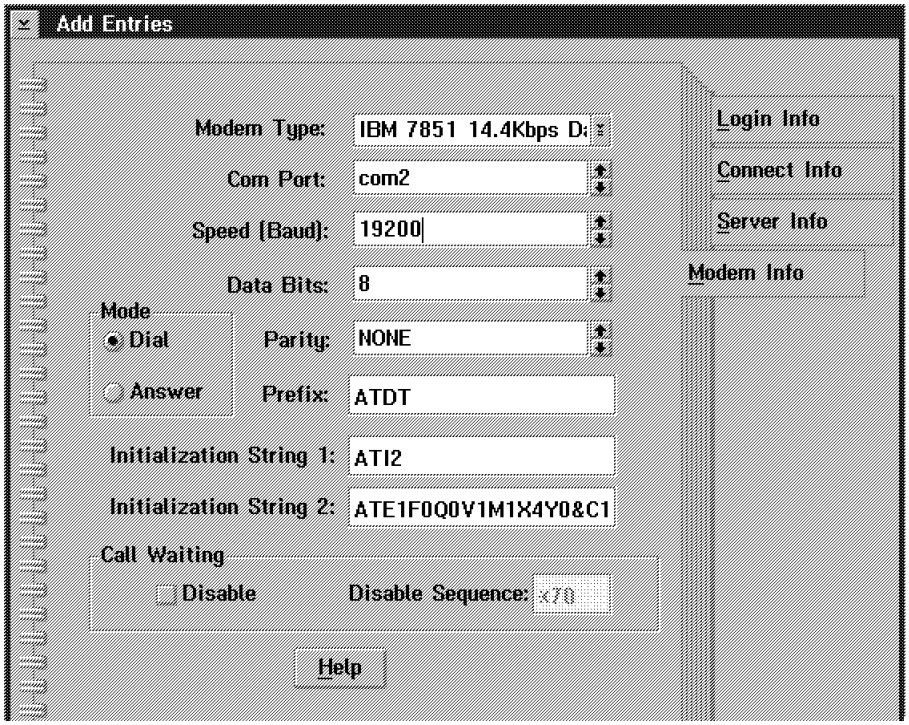


Figure 45. Modem Info

3.3 Point-To-Point (PPP)

PPP is *not* available on the AS/400 system. PPP is briefly discussed here for information purposes only.

3.3.1.1 What is PPP?

PPP stands for Point-to-Point Protocol (RFC1331) and describes a more robust standardized procedure for running TCP/IP over any point-to-point communication line. PPP has the following three components:

PPP Component	Description
Encapsulating	PPP encapsulates the data packets into standard HDLC. The protocol field defines which protocol is being encapsulated. Some examples are Novell′s IPX, TCP/IP, or others.
Link Control Protocol	PPP Link Control Protocol (LCP) is used to establish, control, and disconnect the point-to-point connection.
Network Control Protocol	A PPP Network Control Protocol is defined for every network protocol that is supported by PPP. For IP, this is called IPCP (IP Control Protocol).

PPP is fully defined in Internet RFC1331. Because PPP is so robust, it is very popular right now. Its ability to use multiple protocols and the ease of configuration make this the standard for dial-up connectivity to the Internet.

Chapter 4. E-mail, SMTP, and POP3

In addition to World Wide Web usage, the other applications the Internet is most widely used for is electronic mail (or e-mail). Today it is possible to send a message to someone virtually anywhere in the world. E-mail began on the Internet with the academic community and now has spread to business and personal use as well. Businesses are using e-mail to communicate with other businesses; others are using e-mail to communicate with personal friends or family far away. The Internet has really become the e-mail backbone to the world. All major online providers (such as Compuserve and Prodigy) have gateways from proprietary mailing systems to the Internet. In this way, you can send e-mail between Compuserve and Prodigy (through the Internet) even though Compuserve′s proprietary e-mail is not at all compatible with Prodigy′s.

Corporations and large businesses are doing much the same by providing e-mail gateways between the Internet and local e-mail systems and networks. IBM, if we can use them as an example, runs a large SNA network with mail gateways to the Internet. From a single system point-of-view, an IBMer can handle both SNA and TCP/IP mail.

The AS/400 system provides the same e-mail power by integrating a mail gateway between OfficeVision/400 (and SNADS) and TCP/IP′s Simple Mail Transport Protocol (SMTP). So you, too, can use your AS/400 system to be the mail gateway between your SNA network and your business contacts and friends on the Internet.

SMTP with TCP/IP Connectivity Utilities/400 provides the delivery vehicle to deliver mail on the Internet. Internet standards are defined in RFC′s or Internet Request For Comments. Some of the RFC′s defining SMTP are RFC821 and RFC822. A good source of information regarding RFC′s on SMTP and RFC′s in general is available on the WWW at:

`http://www.cis.ohio-state.edu/hypertext/information/rfc.html`

OfficeVision/400 provides the consistent user interface and seamless access to AS/400 data and facilities so that you really do not care if the people you are sending the note to are located in your local SNA network or on the Internet.

Functions supported by SMTP on the AS/400 system are:

- Distributing the following:
  - AS/400 documents in final-form text format
  - Blind carbon copy on mail distributions
  - Notes and messages using the OfficeVision/400 licensed program
  - Documents and messages
  - Documents and messages using the AS/400 Send Distribution (SNDDST) command and the Receive Distribution (RCVDST) command
  - SMTP notes as an intermediate TCP/IP hop on an SMTP distribution
- Automatic enrollment of the incoming mail′s sender address in the system distribution directory and specified alias tables.
- Personal and system alias table names through SNADS.

- Adding another level of retries for the remote name servers.
- Routing mail to a mail router system if your AS/400 system is unable to deliver the mail.

The AS/400 system has had e-mail capabilities since its introduction. The OfficeVision/400 product along with SNADS allows AS/400 users to e-mail to other AS/400 users. With the addition of TCP/IP Connectivity Utilities/400, the AS/400 system can participate in an SMTP environment. This allows users on an AS/400 system to send mail not only to other AS/400 systems within their own network, but to any SMTP mail server on the Internet. Figure 46 shows the relationship between OfficeVision/400, SNADS, and SMTP. The users all operate in what they think is an SNA environment. They address (even TCP/IP users) with either an OfficeVision/400 nickname or an SNA USERID and address. After the note has been sent, it is up to SNADS and SMTP to make use of the System Distribution Directory to map the SNA USERID and address to that of the Internet.

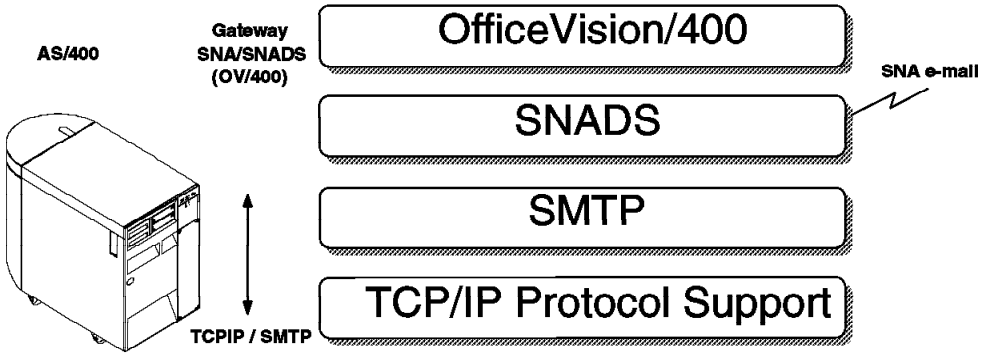


Figure 46. The Layers to an AS/400 E-mail Gateway System

An advantage of using the AS/400 system as the mail server is that users can get the same ″look and feel″ sending or receiving mail whether it be to another AS/400 user or someone on the Internet. Additionally, a user′s mail is saved with the proper system backup procedures in place.

The basic steps involved in setting up SMTP to send and receive mail are:

1. Update the System Distribution Directory.
2. Change SMTP attributes (autostart server and automatic registration).
3. Configure a local domain and host name.
4. Configure to use the remote name server.

Let′s take a look at each of the preceding configuration steps in the next section.

Complete information and specific instructions to configure the preceding steps is available in the *TCP/IP Configuration and Reference*, SC41-3420.

Note: Presently the AS/400 system cannot be a name server; therefore, a remote name server must be defined. SMTP needs a name server to be able to determine addressing information for TCP/IP domain names or host names with fully-qualified domain names. These must be defined in the TCP/IP host table on all systems. Your Internet Service Provider may be able to provide the remote

name serving function for you. There are other alternatives such as a PS/2 running OS/2 and TCP/IP acting as a name server for your network. The Redbook *TCP/IP V2.0 for OS/2 Installation and Interoperability*, GG24-3531, is an excellent reference for setting up an OS/2 name server.

4.1 The Quick Guide to SMTP Configuration

SMTP configuration can be as complex or as simple as you need it to be. The key to a successful configuration is one that masks the end user from the complexity involved in being a gateway between SNADS and SMTP. That is, the end user should see each and every one as either a nickname or a USERID and system, and not really care much if that person is located locally, on another SNA based mail system (or across an intranet or the Internet). So, the ultimate goal is when the user is presented with the Send Note display (see Figure 54 on page 64) and fills in the nicknames or the user ID and address of the addressees and presses Enter that these names all have the configuration behind them to both route to the SMTP gateway and convert to the usually longer more exotic SMTP naming scheme of user@host.domain.

Let's configure our AS/400 system named SYSNM001 to communicate with any user on a remote system named finneous.ibm.com through SMTP in five (easy?!) steps.

4.1.1 Update the System Distribution Directory

An entry is required in the System Distribution Directory for every user that you want to send mail to. This is because the SNADS-to-SMTP gateway must be able to convert the remote users e-mail user@host.domain into an SNA USERID system name in order to route the e-mail through the SNADS and the rest of the SNA network. For remote users, group entries are recommended. A group entry is defined by typing *ANY in the User ID field instead of a specific name. This associates all user IDs with the same address to this entry. Thus, individual entries for each user at that address are not required, and this saves you a lot of administration work in the long run.

For SMTP, one of the simplest approaches is to define one entry for each remote host system. This is done by typing *ANY for the user ID and typing the TCP/IP host name for the address. If a host name is longer than eight characters, a name of eight characters or less should be selected to represent it. This is the usual situation as the remote TCP/IP host name along with the domain information is usually longer than eight characters and can be up to 255 characters! The association to the longer host name is defined by an SMTP alias with the `WRKNAMSMTP *SYSTEM` command.

Most of the configuration that is needed for SMTP is found on the Configure TCP/IP SMTP display (sort of the same as a home page for SMTP) by using the `CFGTCPSMTP` command.

```
                          Configure TCP/IP SMTP
                                                          System:   SYSNM001
Select one of the following:

     1. Work with system alias table
     2. Work with personal alias table
     3. Change SMTP attributes

  SNADS related options:
    10. Work with directory entries
    11. Work with distribution queue for SMTP
    12. Configure distribution services

Selection or command
===> 10
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
(C) COPYRIGHT IBM CORP. 1987, 1994.
```

Figure 47. CFGTCPSMTP Takes You to the Configure TCP/IP SMTP "Home Page"

From the Configure TCP/IP SMTP display, take option 10, Work with directory entries (WRKDIRE or ADDDIRE is faster). We want to take option 1 to Add a directory entry. Add an entry for:

1. Any user on the remote system named Finneous.

```
                          Add Directory Entry

Type choices, press Enter.

  User ID/Address . . . .   *ANY      FINNEOUS
  Description . . . . . .   Any user on the system Finneous (TCP/IP)
  System name/Group . . .   TCPIP     _________     F4 for list
  User profile  . . . . .   __________             F4 for list
  Network user ID . . . .   ____________________________________

  Name:
    Last  . . . . . . . .   ____________________________________
    First . . . . . . . .   ____________________
    Middle  . . . . . . .   ____________________
    Preferred . . . . . .   ____________________
    Full  . . . . . . . .   ____________________________________

  Department  . . . . . .   __________             F4 for list
  Job title . . . . . . .   ____________________________________
  Company . . . . . . . .   ____________________________________
                                                                More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F14=Add X.400 O/R name
F18=Display location details       F19=Add name for SMTP
```

Figure 48. Adding a Directory Entry for Any Remote User on Finneous

2. Any remote TCP/IP SMTP user that originally came into our SNA network on this system. As we see later, we are automatically registering these users in the System Distribution Directory and the SMTP system alias table. This is the entry that allows replies to those remote users to be routed to the SMTP gateway.

```
                              Add Directory Entry

Type choices, press Enter.

  User ID/Address . . . .   *ANY      QSMRMTAD
  Description . . . . . .   Any gateway user that is going back out TCP/IP
  System name/Group . . .   TCPIP     ________      F4 for list
  User profile  . . . . .   __________             F4 for list
  Network user ID . . . .   ____________________________________________

  Name:
    Last  . . . . . . . .   ____________________________________________
    First . . . . . . . .   ____________________
    Middle  . . . . . . .   ____________________
    Preferred . . . . . .   ____________________
    Full  . . . . . . . .   ____________________________________________

  Department  . . . . . .   __________           F4 for list
  Job title . . . . . . .   ____________________________________________
  Company . . . . . . . .   ____________________________________________
                                                                    More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F14=Add X.400 O/R name
F18=Display location details       F19=Add name for SMTP
0 changes made to distribution lists.                                    +
```

Figure 49. Adding a Directory Entry for Any Remote User Gateway

3. A dummy entry needed by SMTP.

```
                              Add Directory Entry

Type choices, press Enter.

  User ID/Address . . . .   QSMTPDMY  QSMTPSYS
  Description . . . . . .   QSMTPQ USER
  System name/Group . . .   TCPIP     ________      F4 for list
  User profile  . . . . .   __________             F4 for list
  Network user ID . . . .   ____________________________________________

  Name:
    Last  . . . . . . . .   ____________________________________________
    First . . . . . . . .   ____________________
    Middle  . . . . . . .   ____________________
    Preferred . . . . . .   ____________________
    Full  . . . . . . . .   ____________________________________________

  Department  . . . . . .   __________           F4 for list
  Job title . . . . . . .   ____________________________________________
  Company . . . . . . . .   ____________________________________________
                                                                    More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F14=Add X.400 O/R name
F18=Display location details       F19=Add name for SMTP
0 changes made to distribution lists.                                    +
```

Figure 50. Adding a Directory Entry for QSMTPDMY (Mandatory)

4.1.2 Change SMTP Attributes - Autostart Server and Automatic Registration

From the Configure TCP/IP SMTP menu, select option 3, Change SMTP attributes. You should see a display similar to Figure 51. Make sure you have Autostart server (*YES) and Automatic Registration (*YES). The defaults here are OK, unless you have a more complex network.

Automatic registration is a good idea for those situations where you cannot plan all of the traffic that flows between your SNA network and the Internet. Since a local user cannot respond to mail without the sender of incoming mail being enrolled in the System Distribution Directory or the alias table, automatic registration is a good tool to have turned on.

About the only problem with automatic registration is that the SNA USERIDs it generates are not too user friendly. Your users see mail from QSM00008 QSMRMTAD, not johndoe@host.domain. In OfficeVision/400, however, users should be encouraged to nickname these ugly gateway USERIDs so that sending mail in the future is as simple as JOHNDOE, for example. Nicknames in OfficeVision/400 are created with option 7 (Directories/distribution lists) from the main menu, then option 4 (Nicknames).

```
                    Change SMTP Attributes (CHGSMTPA)

Type choices, press Enter.

Autostart server . . . . . *YES          *YES, *NO, *SAME
Retries by minute:
  Number of retries  . . . . . 3         0-99, *SAME, *DFT
  Time interval  . . . . . 30            0-99, *SAME, *DFT
Retries by day:
  Number of retries  . . . . . 0         0-9, *SAME, *DFT
  Time interval  . . . . . 0             0-9, *SAME, *DFT
Retry remote name server . . . . *YES    *YES, *NO, *SAME
Automatic registration . . . . . *YES    *NO, *YES, *SAME
  User ID prefix . . . . . QSM           Name, *SAME, *DFT
  Address  . . . . . QSMRMTAD            Name, *SAME, *DFT
  System name  . . . . . TCPIP           Character value, *SAME, *DFT
  Alias table type . . . . . *SYSTEM     *SAME, *SYSTEM, *PERSONAL
User ID delimiter  . . . . . '?'         *SAME, *DFT, ?, =, ., &, $...

                                                                   More...
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 51. Change the SMTP Attributes (CHGSMTPA)

4.1.3 Configure a Local Domain and Host Name

Use the CFGTCP command (another home page for TCP/IP configuration) and select option 12 (Change local domain and host names) to verify that the local domain and host name has been set correctly. SMTP does not work properly if it is not. Please see Figure 52 on page 63 for an example of our system.

If you are running your own Intranet, you are the person to administrate your own domain and host name. If you are connecting to the Internet, then your local ISP provides you with this information.

```
                    Change Local Domain and Host Names
                                                          System:   SYSNM001
Type choices, press Enter.

  Local domain name . . .   SYSNMAA.IBM.COM
                                                                              
                                                                              
                                                                              
  Local host name . . . .   SYSNM001
                                                                              
                                                                      Bottom
F3=Exit   F12=Cancel
```

Figure 52. Local Host and Domain Names are Mandatory for SMTP

4.1.4 Configure the Remote Name Server

Since you probably want to converse with systems all over the Internet, you need a way to convert all of those long host and domain names into something useful such as IP addresses. A remote name server does this for you.

If you are setting up your own Intranet, then you have two choices:

1. For a small network, you can always use the host table support that is available on all TCP/IP systems including the AS/400 system. As systems are added, changed, or removed from the network, you must change this information on all of your systems. That is why we suggest that this is only for small networks.
2. For medium to large networks, you should set up a domain name server on one of your TCP/IP systems. The IP address of that system is the one you should configure here.

   Note: The AS/400 system cannot be a domain name server. We can make use of a domain name server, however, and that is what you are configuring here.

For an AS/400 system that is going to be connected to the Internet, your ISP should provide a domain name server. Get the IP address of this system from your ISP and enter that here.

```
                      Change Remote Name Server
                                                          System:   SYSNM001
Type choices, press Enter.

  Server address  . . . .   9.5.100.76       Internet address
                            9.5.100.75
                                             
  Server port . . . . . .   53               1-65535
  Server protocol . . . .   *UDP             *UDP, *TCP
  Retries . . . . . . . .   3                1-99
  Retry interval  . . . .   2                1-99 (seconds)
  Searched first  . . . .   *REMOTE          *REMOTE, *LOCAL
                                                                      Bottom
F3=Exit   F12=Cancel
```

Figure 53. User Simply Enters Nicknames for the Addressees

4.1.5 Testing Our Simple SMTP Configuration

To test our simple SMTP configuration, let's send a note from our AS/400 system to a user on the Finneous host system. This tests our ability to route SNA based mail out through the SMTP gateway that we just configured.

The second test is for the user (known as Finneous) on the host Finneous to reply back to the AS/400 user. This, then tests our configuration back into the AS/400 system as well as the automatic configuration we configured.

OK, let's get started. On our AS/400 system, the user BRSMITH must be in the System Distribution Directory and must be a registered OfficeVision/400 user. We already have done those steps long before we started with the preceding SMTP configuration. User BRSMITH takes option 4 (Send Note) from the OfficeVision/400 main menu. The user is then presented with the following Send Note display: (Please see Figure 54).

```
                                Send Note

Type mailing information, press F6 to type note.
  Subject  . . . . . .   Hey, Finneous, did you get this?
_____________
  Reference  . . . . . .   ____________________________________________
_____________

Type distribution list and/or addressees, press F10 to send.
  Distribution list  . . . . .   ________ ________   F4 for list

-----Addressees------
User ID      Address      Description
FINNEOUS     FINNEOUS     Any user on the system Finneous (TCP/IP)
________     ________
________     ________
________     ________                                              More...
F3=Exit   F6=Type note   F9=Attach memo slip   F10=Send   F11=Change details
F12=Cancel   F13=Change send instructions   F14=Specify list   F24=More keys
To send, press F10.
```

Figure 54. AS/400 User Enters Subject, User ID, and Address

Notice that the User ID and Address happen to be the same in this case. In most situations, this is not the case. Figure 54 shows us the Description (which is filled in after you press Enter) ″Any user on the system Finneous (TCP/IP)″ that we used for the System Distribution Directory entry of *ANY FINNEOUS. This is a good visual tool to let you know that SNADS is routing this note through the SMTP gateway as you have configured it to do.

We do not show you these steps, but we added some text with PF6 and then sent the e-mail to Finneous using the PF10 key.

Finneous is using OS/2 Warp with UltiMedia Mail Lite Version 2.10 that is installed on the laptop. When the mail arrives, the laptop sings a few notes to let Finneous know that there is some new mail. Here is Finneous's In-basket to show that a new message has been received:

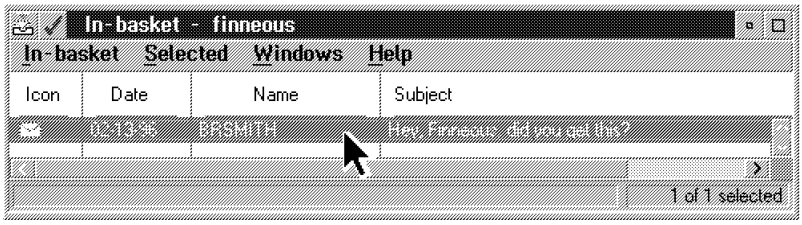


Figure 55. OS/2 UltiMedia Mail/2 In-basket with New Message

Finneous then double-clicks on the new message in the In-basket and brings up the full text of the message that was received from the AS/400 system as shown in Figure 56.

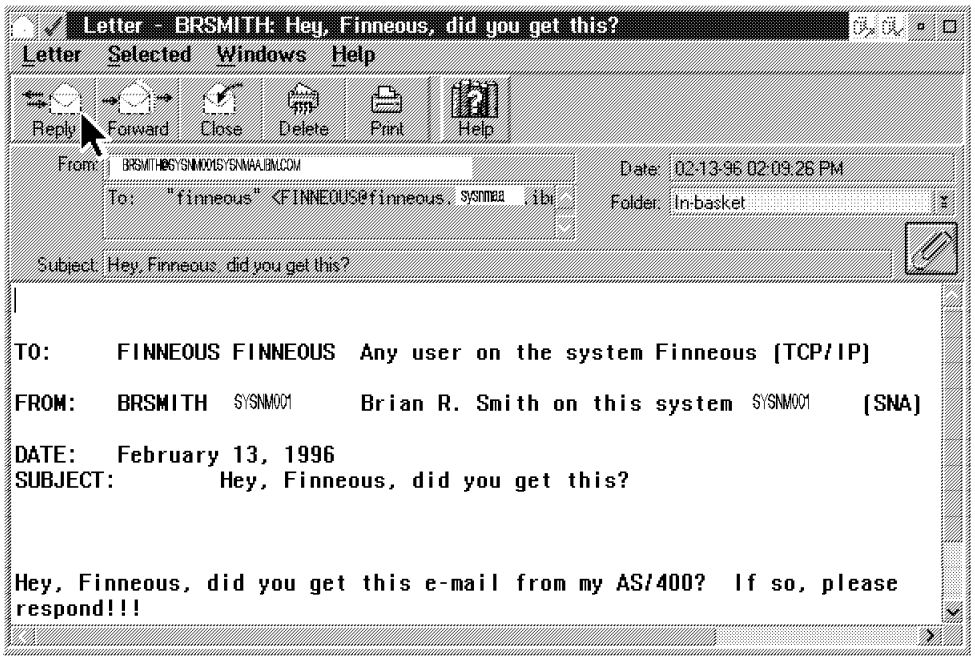


Figure 56. OS/2 UltiMedia Mail/2 Letter from AS/400 System!

As requested, Finneous replies to the message and sends it back to the AS/400 system as shown in Figure 57.

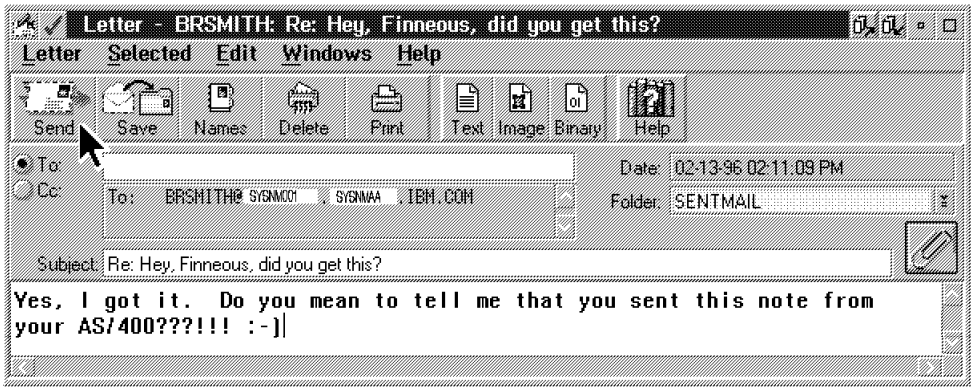


Figure 57. OS/2 UltiMedia Mail/2 Reply to Letter from AS/400 System

Back on the AS/400 system, the user selects Option 2 Mail from the OfficeVision/400 main menu. This takes you to the Work with Mail display where you then use Option 5 to View the new mail as shown in Figure 58 on page 66. Notice that on the Work with Mail Display that is not shown, the "From" User ID and Address is a strange QSM00061 QSMRMTAD. You might ask where did these come from?

```
MAIL         P:12                VIEW Instruction          Pg:1       Ln:1
<2....:....3....:....4....:....5....v....6....:....7....:....8....:....9>...:..

Received: from finneous.sysnmaa.ibm.com by SYSNM001.SYSNMAA.IBM.COM (IBM
Received: by finneous.sysnmaa.ibm.com (IBM OS/2 SENDMAIL VERSION 1.3.14/2
Message-Id: <9602131745.AA0109@finneous.sysnmaa.ibm.com>
Mime-Version: 1.0
Date: Tue, 13 Feb 96 12:44:57
From: finneous@finneous.sysnmaa.ibm.com
To: BRSMITH@SYSNM001.SYSNMAA.IBM.COM
Subject: Re: Hey, Finneous, did you get this?
X-Mailer: Ultimedia Mail/2 Lite, IBM T. J. Watson Research Center
Content-Id: <78_78_1_824233497>
Content-Type: text/plain; charset="US-ASCII"
Content-Transfer-Encoding: 7bit

Yes, I got it.  Do you mean to tell me that you sent this note from
your AS/400???!!! :-)

 F3=Exit           F7=Window       F12=Cancel           F16=File remote
 F4=Find char      F8=Reset        F13=Edit option      F17=Function
 F5=Goto           F10=Forward     F14=Delete mail      F19=Print
 F6=Find           F11=Reply       F15=File local       F21=Nondisplay keys
Text outside of margins cannot be viewed.
```

Figure 58. AS/400 User Selects Option 2, Then 5 to Read the New Mail

The answer lies in the fact that we configured the AS/400 system to automatically register incoming originators of mail on the gateway AS/400 system. The "From" ID of QSM00061 QSMRMTAD is now the SNA name for the remote user finneous@finneous.sysnmaa.ibm.com. To prove this to ourselves, take a look at the system alias table using the WRKNAMSMTP *SYSTEM command:

```
                          Work with Names for SMTP
                                                        System:   SYSNM001
Alias table type . . . . . . . . :   System


Type options, press Enter.
  1=Add   2=Change   4=Remove   5=Display   6=Print

Opt     User ID      Address      SMTP Name

 _      QSM00061     QSMRMTAD     finneous@FINNEOUS.SYSNMAA.IBM.

                                                                      Bottom
F3=Exit   F5=Refresh   F12=Cancel   F15=Print list   F17=Position to
(C) COPYRIGHT IBM CORP. 1987, 1994.
```

Figure 59. AS/400 Work With Names for SMTP Shows SNA Alias for Finneous

Since the name QSM00061 QSMRMTAD is not such an easy name to remember (unless you are a computer), our AS/400 user can simply use Option 7 Directories/distribution lists on the OfficeVision/400 main menu to create a nickname (option 4=Nicknames) for the remote friend Finneous. Try this for yourself to verify that a new piece of mail sent to the nickname FINNEOUS from the AS/400 system does indeed make it all the way to the SMTP based mailer UltiMedia Mail/2.

4.2 The AS/400 System as an E-mail Gateway

The AS/400 system, in addition to being a Internet SMTP mail server, can be an SMTP/SNADS gateway for a network of AS/400 systems. SMTP Mail can be received by a focal point AS/400 system which is then passed to other AS/400 systems that may not have TCP/IP installed. In this scenario, anyone in a corporate SNA network may have access to Internet mail. This is accomplished through the configuration of SNADS.

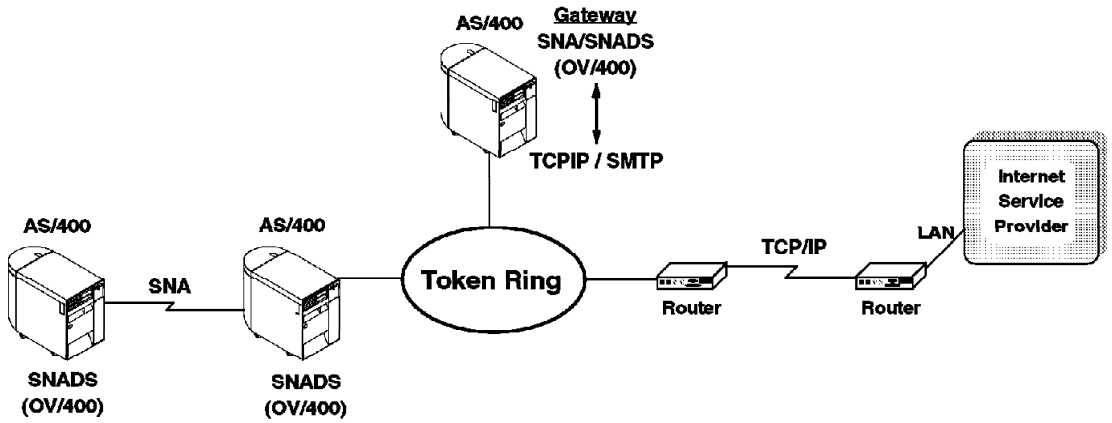


Figure 60. AS/400 SMTP to SNADS Mail Gateway

4.2.1 Sending a Message Without OfficeVision/400 or JustMail/400

Mail can be sent without using the OfficeVision/400 or JustMail/400 licensed programs by using the Send Distribution (SNDDST) command. This command does not support notes. Figure 61 on page 68 shows the Send Distribution (SNDDST) display. In this example, a message is being sent.

```
                        Send Distribution (SNDDST)

Type choices, press Enter.

Information to be sent . . . . . > *MSG          *MSG, *DOC, *FILE, *IDP...
Recipient:                          _
  User ID  . . . . . . . . . . . . > JOHNP        Character value
  Address  . . . . . . . . . . . . > AS400        Character value
  Recipient type . . . . . . . . .  *PRI          Character value, *PRI, *CC...
              + for more values _
Description  . . . . . . . . . .  Message 1
_____
Message  . . . . . . . . . . . .  Text Message using SNDDST
_____________________________________________________________________________
_____________________________________________________________________________
_____________________________________________________________________________
Confirmation of delivery . . . .  *NO            *NO, *YES
Sensitivity  . . . . . . . . . .  *NONE
Content importance . . . . . . .  *NORMAL       *NORMAL, *LOW, *HIGH
Priority . . . . . . . . . . . .  *NORMAL       *NORMAL, *HIGH, *LOW
                                                                      More...
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display        F24=More keys
```

Figure 61. AS/400 SNDDST Display

Note: You must be enrolled in the system distribution directory to use the SNDDST command.

To send a message, do the following:

1. Type SNDDST.
2. Press F4 (Prompt).
3. Type the information requested on the Send Distribution (SNDDST) display.
4. Press the Enter key.

4.2.2 Receiving a Message Without OfficeVision/400 or JustMail/400

Mail can be received without using the OfficeVision/400 or JustMail/400 licensed programs by using the Receive Distribution (RCVDST) command. You must perform the following steps to receive a message if you are not using one of the Office products:

1. Run the Query distribution (QRYDST) command to determine if there is any incoming mail.
2. If there is incoming mail, run the QRYDST command again, this time specifying an output file.
3. Display the contents of the output file, using the Display Physical File Member (DSPPFM) command, to determine the distribution ID.
4. Run the Receive Distribution (RCVDST) command, specifying the distribution identifier to receive the mail.

   Note: You must be enrolled in the system distribution directory to use the Query Distribution (QRYDST) command and the Receive Distribution (RCVDST) command.

4.3 POP3 (Post Office Protocol)

POP3 (the current POP standard) is a standard mail interface that is supported by AIX, Windows, OS/2, and Macintosh clients. Anyone with a POP3 client is then able to define their AS/400 system as a POP3 server. The POP3 server is a simple store and forward mail system. It provides mailboxes for the user′s mail that are periodically queried by the client. The client retrieves the mail from the mailbox on each query and asks the server to delete it from the mailbox. The AS/400 system uses the AnyMail/400 mail server framework and the system distribution directory to process and distribute E-mail.

Here is a picture (please see Figure 62 on page 70) that shows us a few things about this new POP3 server on the AS/400 system.

- The POP3 server on the AS/400 system is not part of SMTP or SNADS. It is a server all on its own listening for POP3 clients on the TCP well-known port of 110. SMTP, for example, listens on TCP port 25.
- The POP3 server, similar to all of the other AS/400 mail components such as OfficeVision/400, SNADS, and SMTP, uses the System Distribution Directory as the focal point for the mapping of names from SNA based mailing to TCP/IP.

  Not shown in the picture is the AS/400 system that is also a mail gateway for OSI′s X.400. The System Distribution Directory also is the database where SNA names are mapped to X.400′s Originator and Recipient (OR) Names.
- The POP3 server, the same as all of the other mail components, now uses the AS/400 AnyMail/400 Mail Framework to help route mail within the system. The AnyMail/400 Mail Framework is a series of programming exit points where code can be registered (or ″snapped″ in) to handle mail requests other than the traditional SNADS. For example (and a real good example), an AnyMail/400 snap-in can be built that is able to handle MIME (Multipurpose Internet Mail Extensions, see 4.3.1, “MIME” on page 71) attachments.

  The AnyMail/400 Mail Framework is written such that anyone can provide a new mail service on the AS/400 system. IBM has done this by providing a POP3 server.
- As shown in the diagram, a connection exists between all of the AS/400 mail components to provide a transfer of mail items.

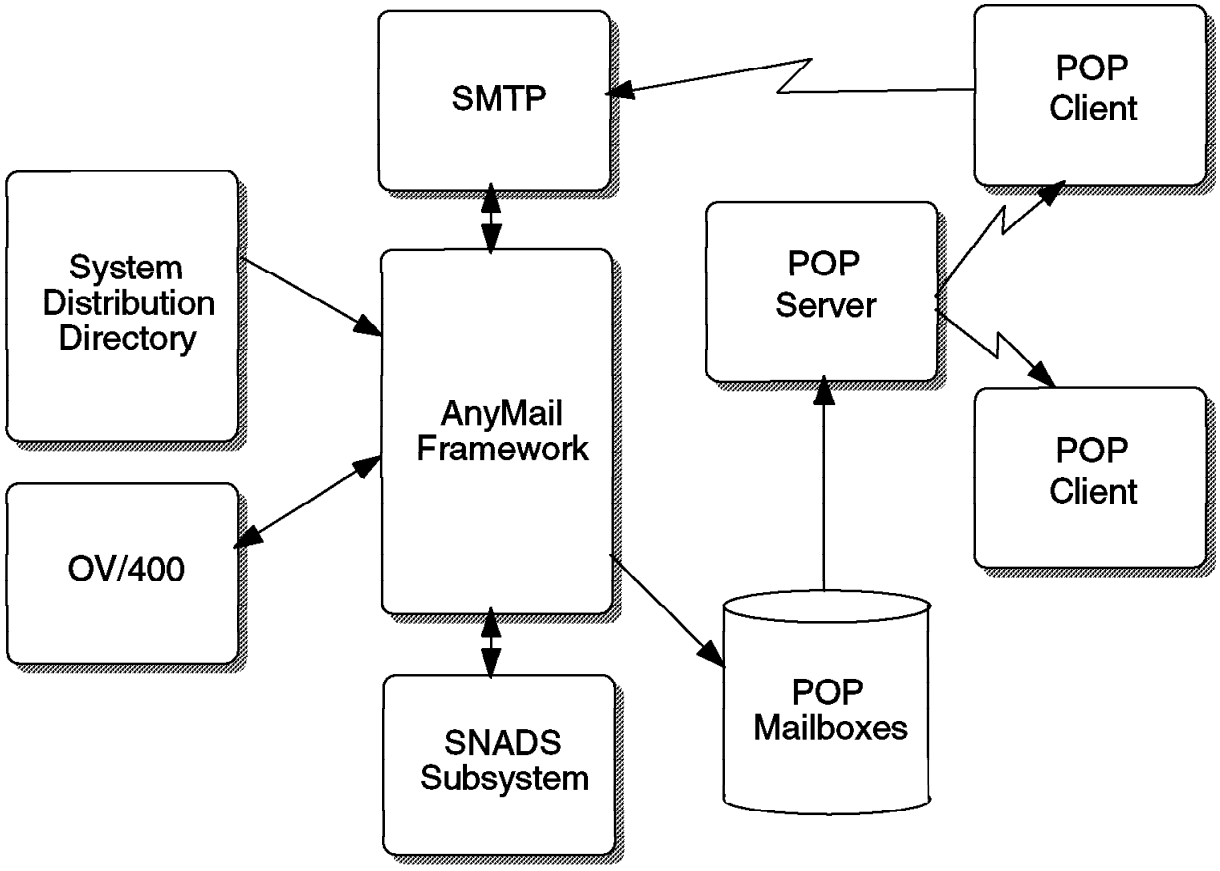


Figure 62. POPS Server Interactions

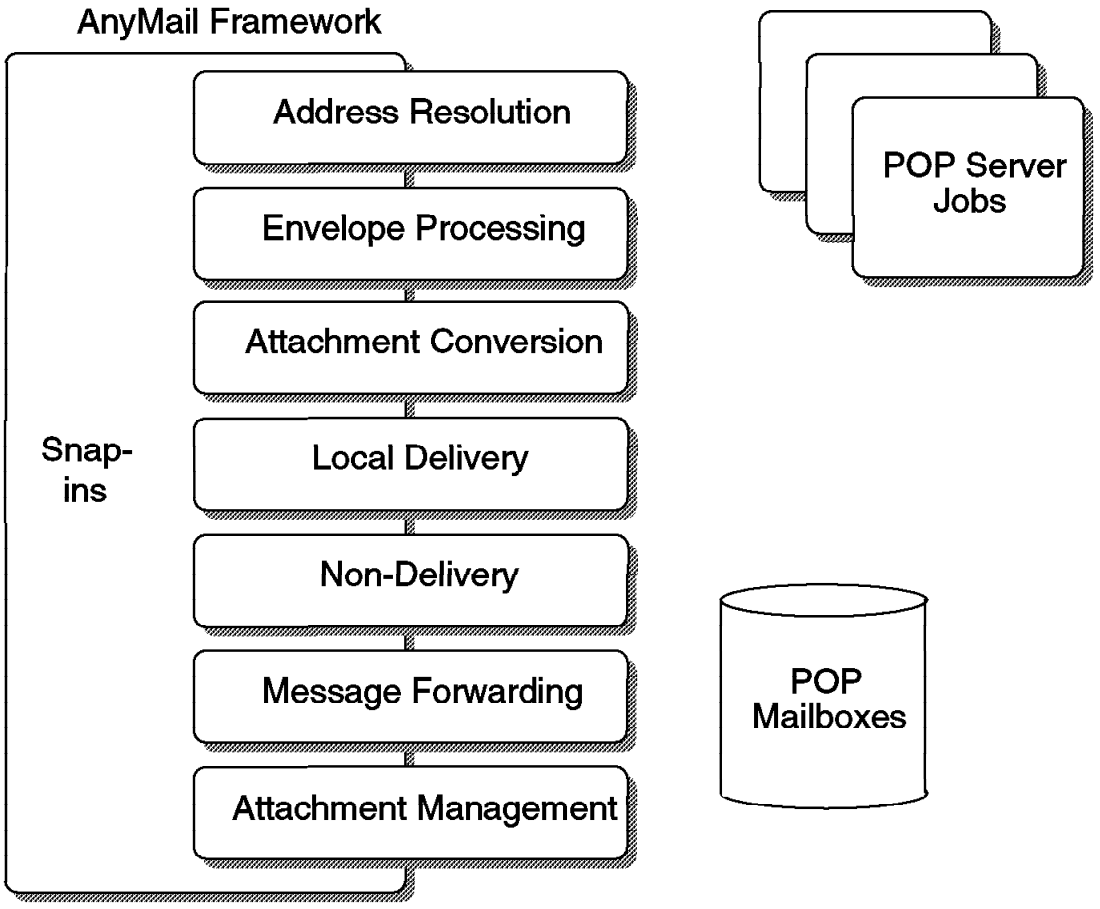


Figure 63. Components of the POP3 Server

Figure 63 shows the components of the POP3 server. Listed are the snap-ins that are used by the POP3 server.

4.3.1 MIME

MIME (Multipurpose Internet Mail Extensions) is a standardized method for organizing divergent file formats. The method organizes file formats according to the file′s MIME type. When a MIME enabled browser such as Netscape retrieves a file from a server, the server provides the MIME type of the file. The browser uses the MIME type to establish whether the file format can be read by the software′s built-in capabilities or, if not, whether a suitable helper application is available to read the file.

For servers that do not provide a MIME type with a file, the browser interprets the file′s extension (a suffix appended to a file name). For example, the .html extension in the file name index.html suggests a file in the HTML format. Similarly, a .zip extension suggests a compressed file, an .rtf extension suggests a file in the Rich Text Format, and so on. You can view and configure the mapping of all MIME types to helper applications. In Netscape, for example, these can be configured by using the Helper Applications display. The helper applications can be simple applications such as PKUNZIP to de-compress a file,

or a PMVIEW to view a JPEG (picture file) in OS/2. Other examples of MIME imbedded information might be sounds or even moving pictures.

OfficeVision/400 allows customers to do an orderly migration from ″green screen″ to workstation-based mail clients. A MIME parsing function has been developed so the OfficeVision is able to receive mail from POP3 clients. This also allows OfficeVision to receive MIME mail from the Internet. OfficeVision (actually a SNADS limitation) can only accept a single attachment with each message whereas MIME supports multiple attachments with a message. Therefore, a single MIME message may result in multiple messages when it is parsed and converted to SNADS formats.

When the POP3 server receives mail for a LAN-attached POP client with the appropriate MIME software, the AS/400 system ensures that the proper MIME headers are retained.

4.3.2 AS/400 POP3 Configuration

As we saw in Figure 62 on page 70, the AS/400 System Distribution Directory plays a very important role in the configuration of OfficeVision/400, SNADS, SMTP, and now the POP3 server. We see just how important as we review the highlights of configuring the AS/400 system as a POP3 server.

1. Use CRTUSRPRF to create the POP3 client to be able to pass security (that is, login to the AS/400 system).
   - For security reasons, you may want to set up these user profiles with parameter INLMNU(*SIGNOFF), since POP3 mail users do not actually sign on to the AS/400 system.

     Note: If you use *SIGNOFF for the initial menu the remote user could still access the AS/400 system to change passwords, display messages and do other things based upon the AS/400 Operational Assist menu. This is because they can press the ATTN key at the moment they saw the Initial program ended and *SIGNOFF specified for initial menu message. One of the PF keys on this menu is F9=Command line which is useful for the remote trusted user, but very dangerous if the userid and password fell into the hands of a vandal.
   - A better solution for the more security conscience administrator would be code code a small CL program as the user profile′s Initial program to call (INLPGM). This small (untested) CL program would do nothing but immediately signoff and end the connection to the telnet client.

     ```
     PGM
     MONMSG CPF0000
     SIGNOFF LOG(*LIST) ENDCNN(*YES)
     ENDPGM
     ```

     We also would want to have a joblog produced so we could be monitoring if this is happening with any great regularity that might alarm us to the fact that a vandal is trying to get in using a stolen userid and password.

     Note: The system administrator will need to manage password expiration, since this user will not know when their password expires.
2. Use the following procedure to add an entry to the system distribution directory.

- Type WRKDIRE. Press Enter. This shows the Work with Directory Entries display.
- Type option **1** to add a directory entry. Press Enter.

Add the preceding user profile to the system distribution directory. This is a local user on this system. The key to this is the Mail service level that specifies the System message store and the Preferred address that specifies the SMTP name. See Figure 64 for an example.

```
Mail service level  . .   2                    1=User index
                                                2=System message store
                                                3=Other mail service
  For choice 3=Other mail service:
    Field name  . . . .                         F4 for list

Preferred address . . .   3                    1=User ID/Address
                                                2=O/R name
                                                3=SMTP name
                                                4=Other preferred address
```

Figure 64. POP3 Configuration Extras in the ADDDIRE Displays

While in the ADDDIRE, you also must add a name for SMTP using the F19 key. This adds an SMTP name for this user profile directly into the System Distribution Directory.

Important!

Because the address you set up on your client is used for your user profile on the AS/400 system that is handling your mail, you need to use the same name for your user profile and your SMTP user ID. IBM also recommends that this name be the same as your system distribution directory user ID, although it is not required.

3. Use CHGPOPA to change the AS/400 POP3 server attributes.

   This is not really a necessary step, but you might want to change some attributes. The defaults for the POP3 server on the AS/400 system are quite acceptable to get most clients running the first time. The Address Book is not available at this time. The following example shows the CHGPOPA display:

```
                    Change POP Server Attributes (CHGPOPA)

Type choices, press Enter.

Autostart servers  . . . . . .   *YES          *YES, *NO, *SAME
Number of initial servers  . . .   3            1-20, *SAME, *DFT
Inactivity timeout . . . . . .   600            10-65535 seconds, *SAME, *DFT
Message split size . . . . . .   128            32-2048 kilobytes, *SAME, *DFT
MIME CCSID:
  Coded character set identifier   00819        *SAME, *DFT, 00819, 00912...
  When to use  . . . . . .   *BESTFIT      *SAME, *BESTFIT, *ALWAYS
Allow standard POP connection  .   *YES          *SAME, *YES, *NO
Host server connection . . . . .   *NONE         *SAME, *NONE, *ALL, *IP...
              + for more values
Address book:
  Enabled  . . . . . .   *NO           *SAME, *NO, *YES
  Refresh interval . . . . . .                  1-65535 minutes, *NONE

                                                                      Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 65. Change POP Server Attributes (CHGPOPA)

4. Start all of the mail servers (if not already started):
   - Use STRSBS QSNADS to start the SNADS subsystem.
   - Use STRMSF to start the Mail Server Framework.

     It does not happen often, but sometimes a mail item is received on the AS/400 system that causes one or more of the MSF jobs to fail during processing. In this event, you can pick a mail message option (MSGOPT on the STRMSF command) of something other than the default of *RESUME. In some situations, the failing MSF job log indicates a software problem and you must use the *CLEAR option that does this:

     ```
     All existing mail server framework messages are
     deleted.  This option should only be used when a
     software error is reported with the mail server
     framework or its associated exit point programs.
     ```

   - Use STRTCPSVR SERVER(*POP) to start the POP3 server.

One of the parameters configured on the client can be to specify how often to query the POP3 server to see if new mail has arrived. On the UltiMedia Mail/2 client, for example, the default is one minute. If new mail (from whatever source) arrives for the user profile on the AS/400 system, the POP3 client knows about it, at most, one minute later.

If the POP3 client was disconnected from the network, which is one of the main reasons for the Post Office Protocol, then the mail sits on the AS/400 system in the Integrated File System until some point in the future when the POP3 client is back online.

4.3.3 Configuration for UltiMail Lite POP3 Client

- Double-click on **TCP/IP**.
- Double-click on **TCP/IP Configuration**.
- Click on **Mail**.

Figure 66 shows Page 1 of Mail Configuration.

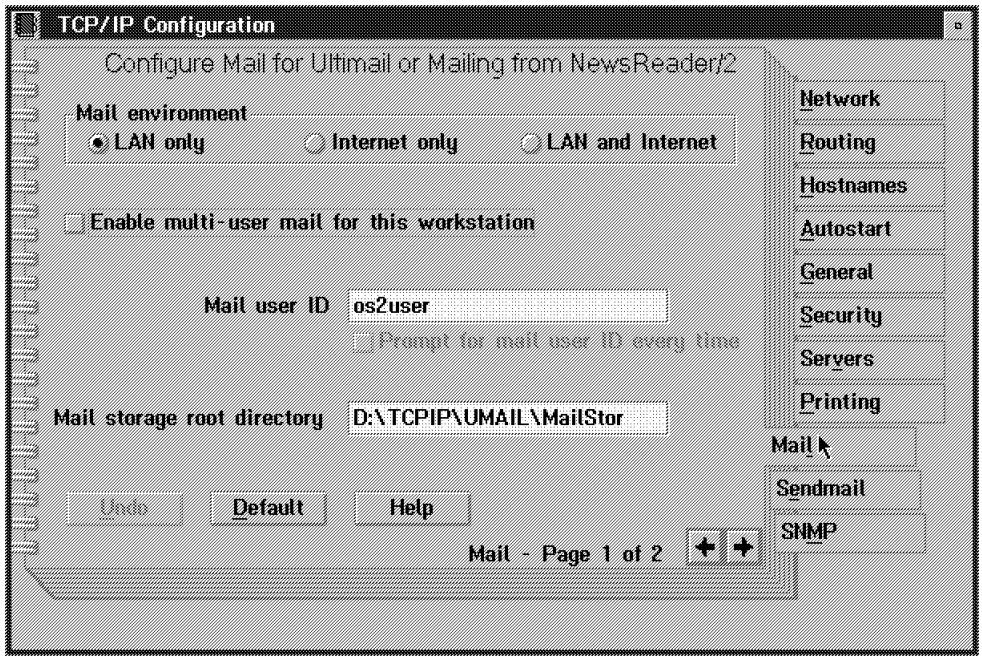


Figure 66. Mail - Page 1 of 2

We filled in the **Mail user ID**. This is who we are known as when we send mail.

- We then clicked on the right arrow to get to **Mail - Page 2 of 2**. Figure 67 on page 76 shows this display:

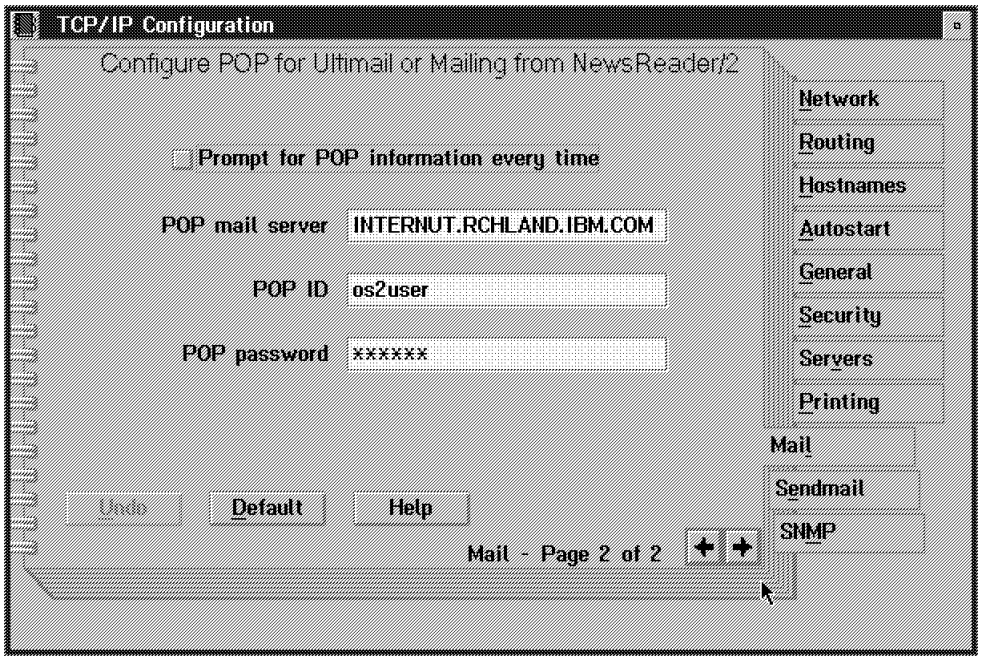


Figure 67. Mail - Page 2 of 2

This display is where we entered our POP mail server (this was our AS/400 system), the POP ID (this was our user profile name on the AS/400 system), and the POP password (this was our AS/400 user profile password). The POP password is not displayed.

- We then clicked on **Sendmail**. Figure 68 shows this window:

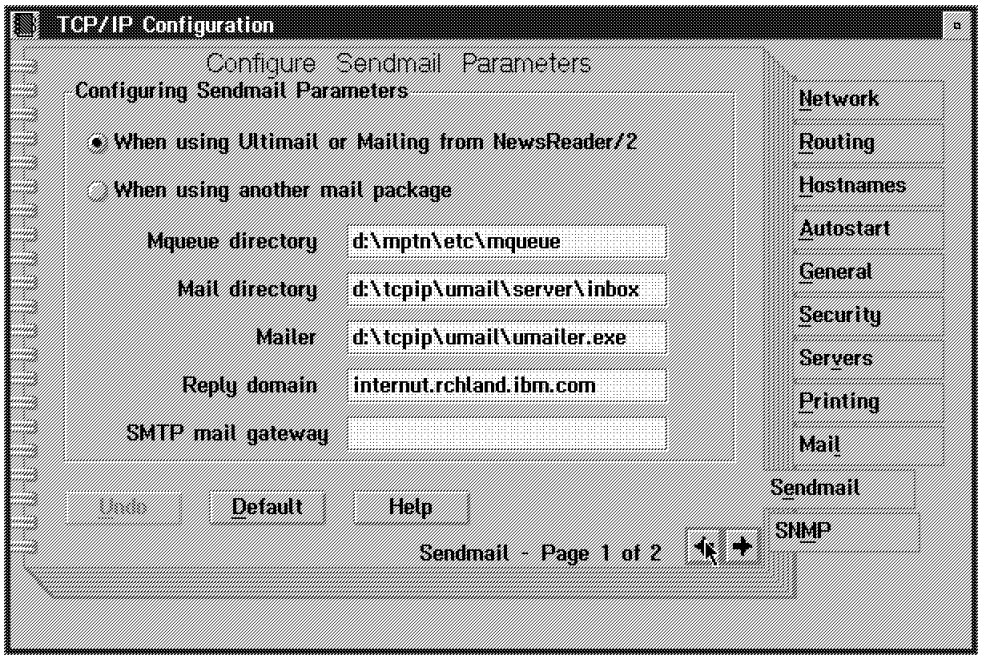


Figure 68. Sendmail - Page 1 of 2

Note: It was necessary to fill in the **Reply domain** parameter to get it to work.

4.3.4 Configuration for Netscape POP3 Client

- Start Netscape.
- Click on **Options** to get the pull-down menu.
- Click on **Mail and News Preferences**.

The **Servers** tab is displayed as shown in Figure 69.

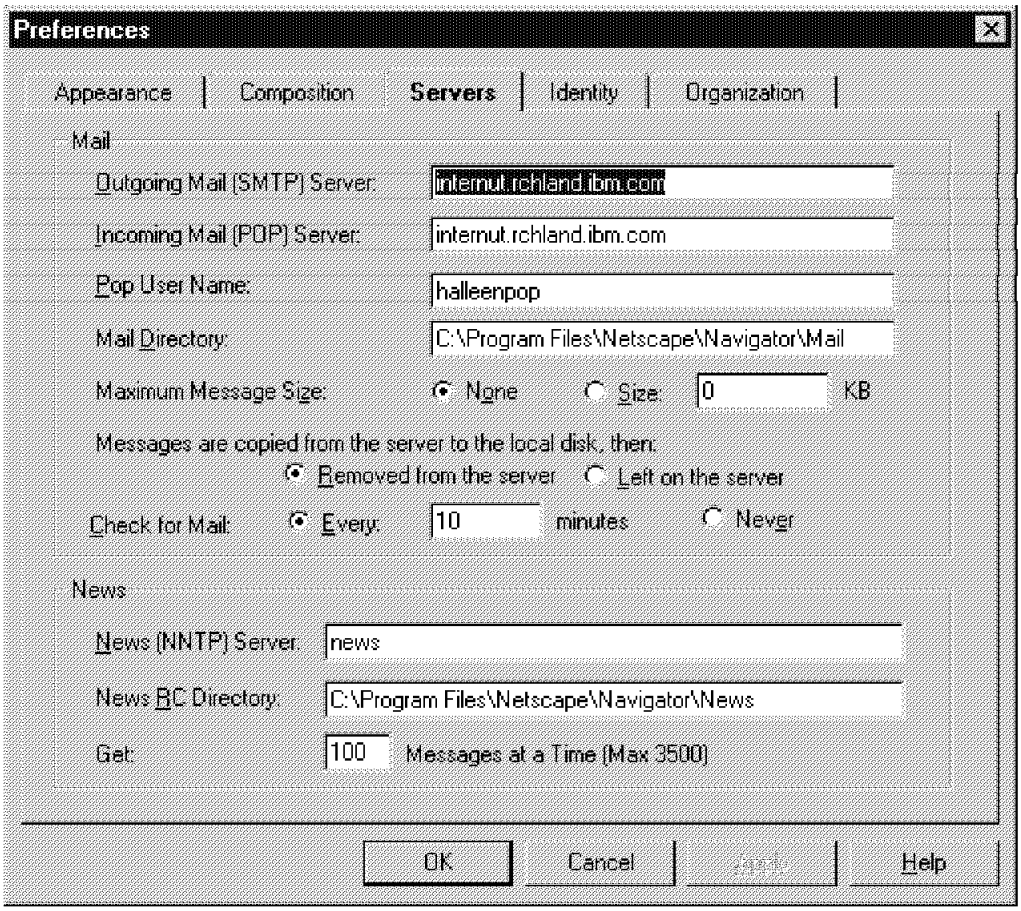


Figure 69. Netscape Mail Servers

This window shows the Outgoing Mail (SMTP) Server and Incoming Mail (POP) Server that were both pointing to my AS/400 system. The POP User Name was listed and this needed to match the local user profile that we created on the AS/400 system for my POP mailbox.

- Click on **Identity** to display the window listed in Figure 70 on page 78.

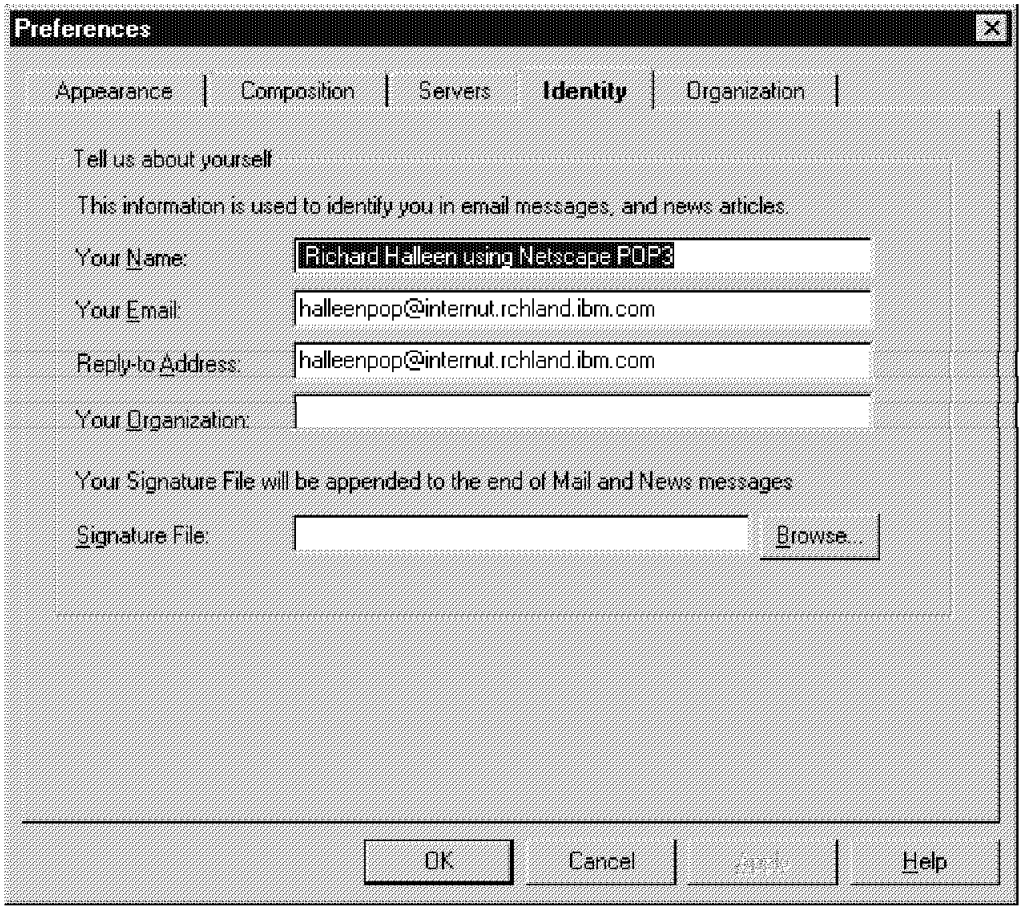


Figure 70. Netscape Mail Identity

This window shows the information used to identify ourselves in e-mail messages that we send.

4.3.5 POP3 Client-to-Client Example

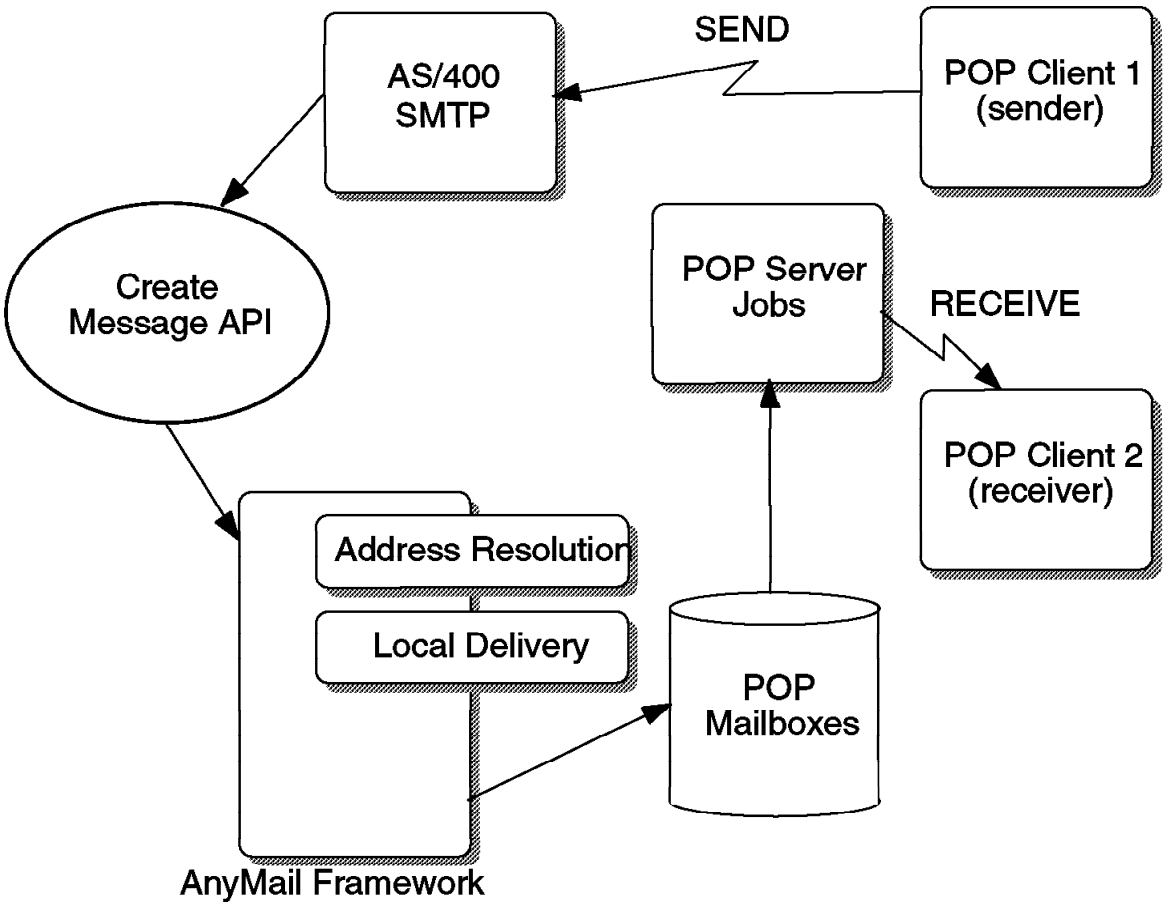


Figure 71. Data Flow - Local/Remote POP3-to-POP3

The following example shows a POP3 client using OS/2 WARP UltiMail Lite sending a piece of mail to a POP3 client using Netscape Mail.

- The UltiMail Lite client creates a new letter as shown in Figure 72 on page 80.

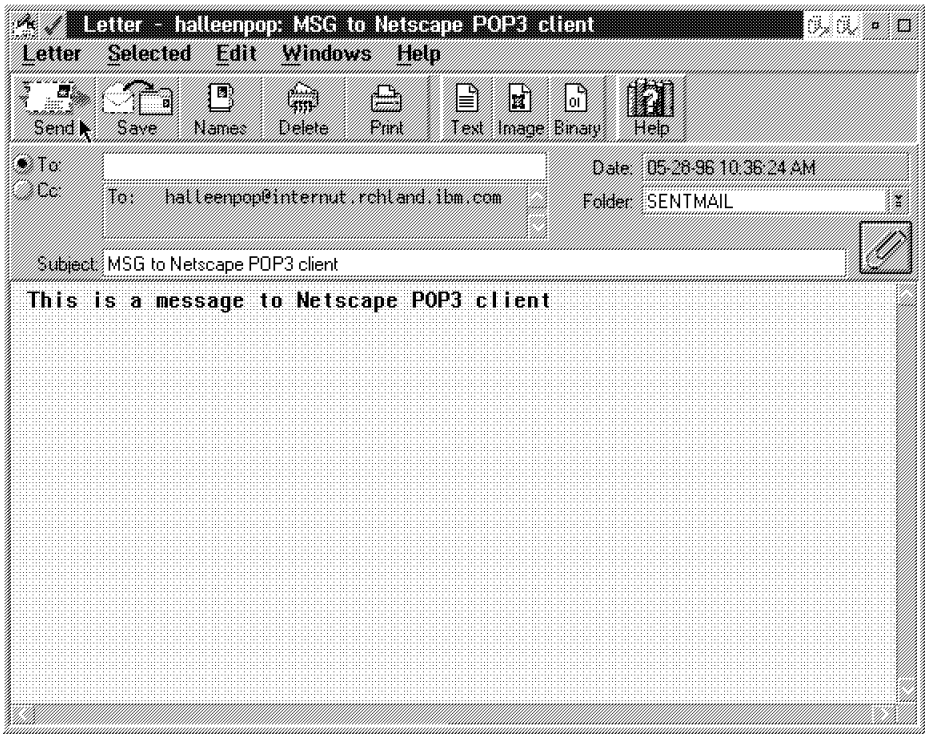


Figure 72. UltiMail Lite Client Sending Mail

- The **To:** field is a POP3 user on our AS/400 system.
- The client then clicks on **Send** to send the mail.
- The mail from the POP3 client is always sent out using SMTP.

The mail sent to the POP3 client is stored in the Integrated File System (IFS) on the AS/400 system before it is received by the POP3 client. It is possible to see if there is mail in a POP3 mailbox, but the specific mail cannot be viewed.

We entered the command `WRKLNK OBJ('QTCPTMM/MAIL/HALLEENPOP/*')` to view if there was mail in our mailbox. The syntax in this command is not case sensitive. Figure 73 shows my mail stored in the IFS.

```
                          Work with Object Links

Directory  . . . . : /qtcptmm/mail/halleenpop

 Type options, press Enter.
   3=Copy  4=Remove  5=Next level  7=Rename  8=Display attributes
   11=Change current directory ...

Opt Object link          Type   Attribute  Text
     JW441735.NOT        STMF

                                                                      Bottom
Parameters or command
===>
 F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F12=Cancel  F17=Position to
 F22=Display entire field          F23=More options
```

Figure 73. Work with Object Links

The Netscape Mail POP3 client then clicks on **Get Mail** or waits until the POP3 client checks for mail automatically based on the client configuration.

Figure 74 shows the new mail:

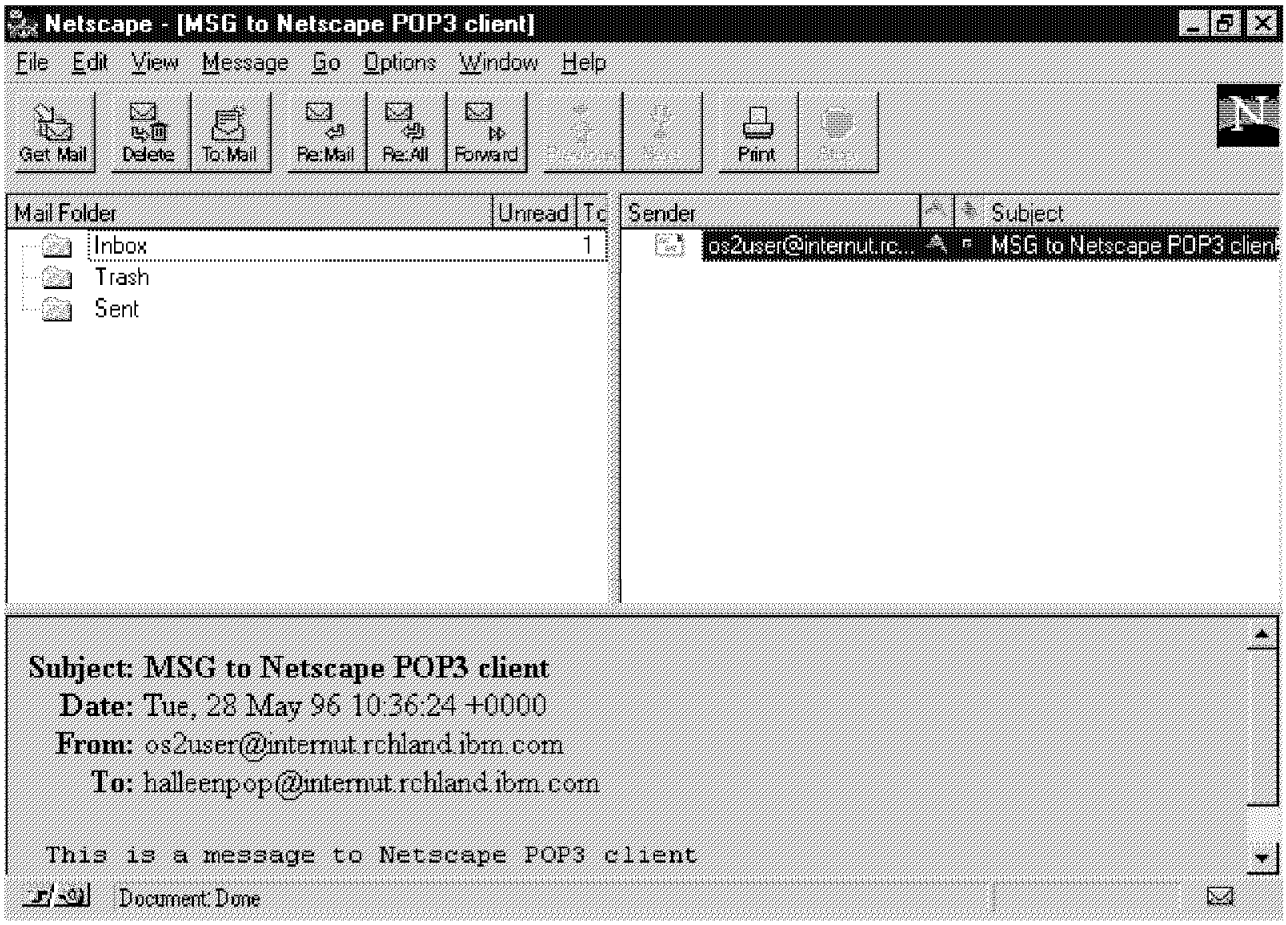


Figure 74. Netscape Mail POP3 Client Viewing the Mail

4.3.6 Example of POP3 Client Sending Mail to OV/400 User with GIF

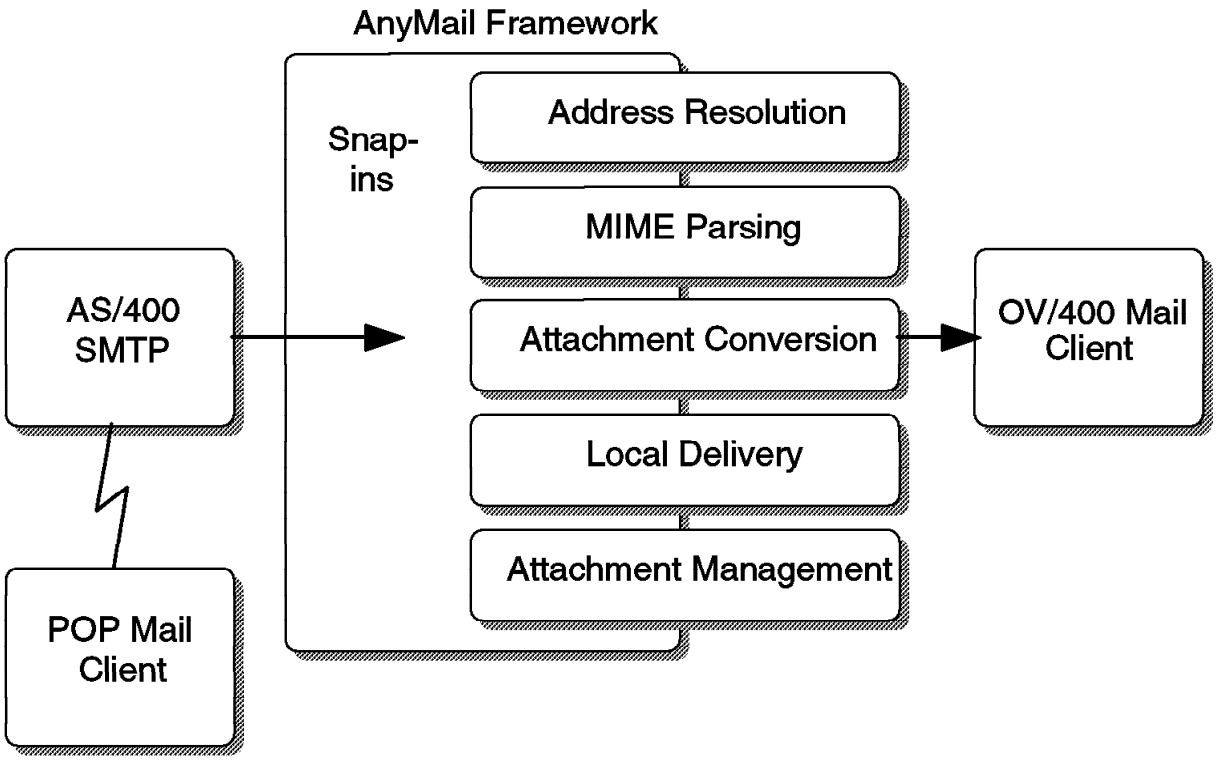


Figure 75. Data Flow Local/Remote POP3 Client to OV/400 Client

The following example shows a POP3 client sending a piece of mail (with a GIF file attached) to an OV/400 user.

Figure 76 on page 83 shows the mail before it is sent from Netscape:

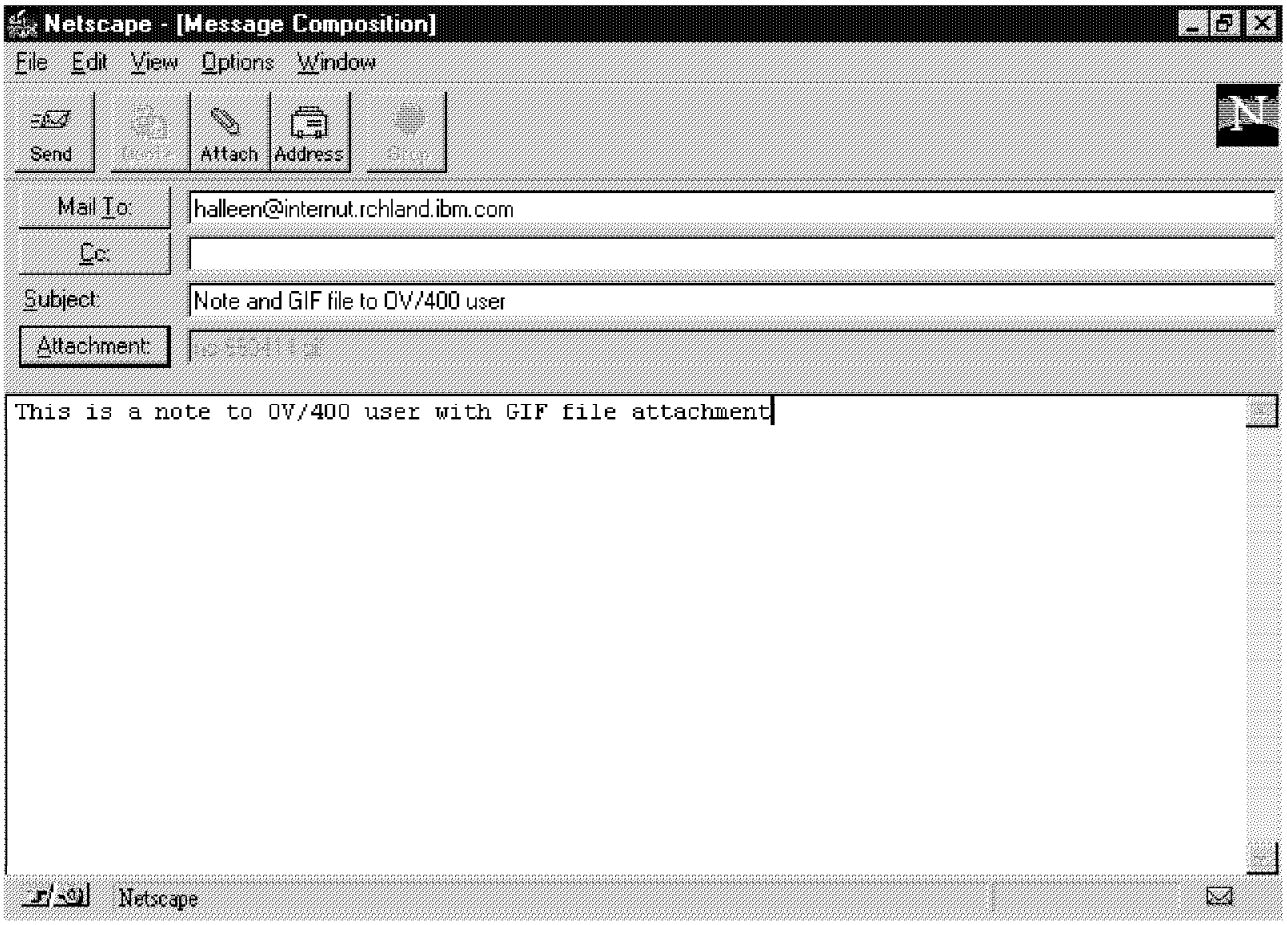


Figure 76. Netscape Mail Message Composition Window

The user clicks on **Send** to send the mail.

Figure 77 on page 84 shows a copy of the mail that was sent (notice the GIF file).

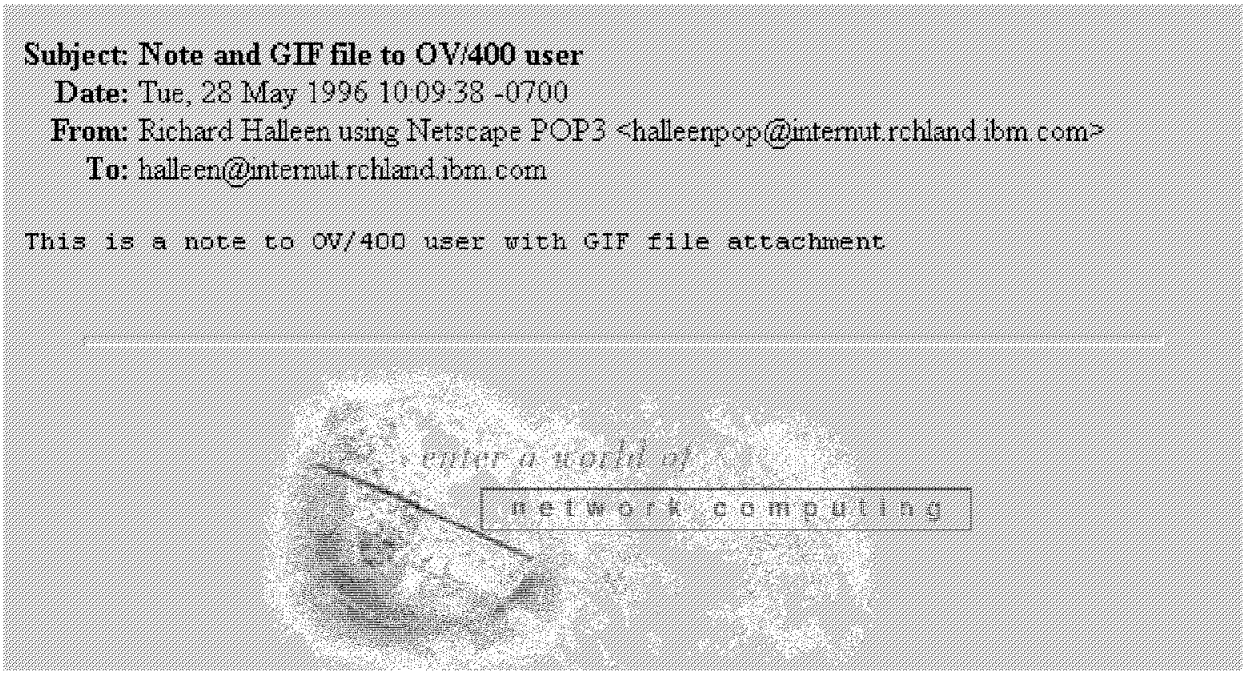


Figure 77. Sent Mail as Displayed in Netscape Mail

Back on the AS/400 system, we go into OV/400 and work with mail for HALLEEN. Figure 78 shows the new OV/400 mail.

```
                                Work with Mail

Working with mail for  . . . . . : HALLEEN INTERNUT


 Type options, press Enter.
   2=Revise a copy  4=Delete    5=View    6=Print  8=Work with details
   9=Print options  10=Forward  11=Reply  12=File remote  13=File local
   14=Authority     15=Fill form

                 -------From-------                                  Date
OptStatus    User ID Address Description                       Received
 5  NEW       HALLEENPINTERNUTNote and GIF file to OV/400 use05/28/96
    NEW       HALLEENPINTERNUT(c-960414.gif)                   05/28/96

                                                                        Bottom
 F3=Exit  F5=Refresh  F6=Work with outgoing mail status
 F10=Display new mail  F11=Change user  F12=Cancel  F24=More keys
```

Figure 78. OV/400 Work with Mail

Notice the two mail entries in Figure 78. One is for the text and one is for the GIF file. OV/400 is *not* a MIME-compliant client. The MIME conversion that is supplied by the mail server framework allows users to view the text portions of MIME notes. The OV/400 user may be able to view the binary portions of the MIME notes if they are using a graphical user interface such as IBM Current OfficeVision/400. However, with only OV/400, binary attachments can be forwarded to MIME-compliant mail users who can display them. They can also be copied to local folders and accessed by using Client Access network drives.

We selected **Option 5** to display the text portion of the note. Figure 79 on page 85 shows the text version of the note.

```
   MAIL        P:12                VIEW                     Pg:1       Ln:16
    <2....:....3....:....4....:....5....v....6....:....7....:....8....:....9>...
:..
   Message-ID:<31AB3352.445C@internut.rchland.ibm.com>
   Date: Tue, 28 May 1996 10:09:38 -0700
   From:  Richard Halleen using Netscape POP3 <halleenpop@internut.rchland.i
   MIME-Version: 1.0
   To: halleen@internut.rchland.ibm.com
   Subject: Note and GIF file to OV/400 user


   ------------------------------------------------------------------------
   This is a note to OV/400 user with GIF file attachment


   ------------------------------------------------------------------------
   Type/Subtype: image/gif
   Description: (c-960414.gif)
    ____________________________________________________________________________
__   F3=Exit           F7=Window       F12=Cancel           F16=File remote
     F4=Find char      F8=Reset        F13=Edit option      F17=Function
     F5=Goto           F10=Forward     F14=Delete mail      F19=Print
     F6=Find           F11=Reply       F15=File local       F21=Nondisplay keys
   Text outside of margins cannot be viewed.
```

Figure 79. OV/400 View of Text Note

Notice the **Type/Subtype** in Figure 79. The Type is image and the Subtype is gif. This is mapped to OV/400 as a PC file and is listed as a separate mail item as shown in the previous figure.

We go back to the OV/400 **Work with Mail** display as listed in Figure 80.

```
                              Work with Mail

Working with mail for  . . . . . : HALLEEN INTERNUT


 Type options, press Enter.
   2=Revise a copy  4=Delete    5=View    6=Print  8=Work with details
   9=Print options  10=Forward  11=Reply  12=File remote  13=File local
   14=Authority     15=Fill form

                -------From-------                                  Date
OptStatus    User ID Address Description                     Received
    OPENED    HALLEENPINTERNUTNote and GIF file to OV/400 use05/28/96
 13 NEW       HALLEENPINTERNUT(c-960414.gif)                  05/28/96

                                                                      Bottom
 F3=Exit  F5=Refresh  F6=Work with outgoing mail status
 F10=Display new mail  F11=Change user  F12=Cancel  F24=More keys
```

Figure 80. Option 13 - File local

We entered option 13 to file the GIF file in a local folder. Figure 81 on page 86 shows the display.

```
                              File Document

Document type . . . . . : IBM personal computer file

 Type choices, press Enter.

  Document  . . . . . .  c-960414.gif         Name, *NONE
  Folder  . . . . . . .  HALLEEN
                                                Name, *NONE, F4 for list
  Document description     Note and GIF file to OV/400 user
  Subject . . . . . . .  (c-960414.gif)

  Authors . . . . . . .
  Keywords  . . . . . .

                                                F4 for list
  Document class  . . . .                      F4 for list
  Allow revisions . . . .  Y                   Y=Yes, N=No
  Delete from mail  . . .  N                   Y=Yes, N=No
  Project . . . . . . .
                                                                    More...
 F3=Exit  F4=Prompt  F5=Refresh  F9=Change authority  F12=Cancel
 F19=Display messages
```

Figure 81. File Document

4.3.7 Example of OV/400 Forwarding MIME Binary Attachment to POP3 Client

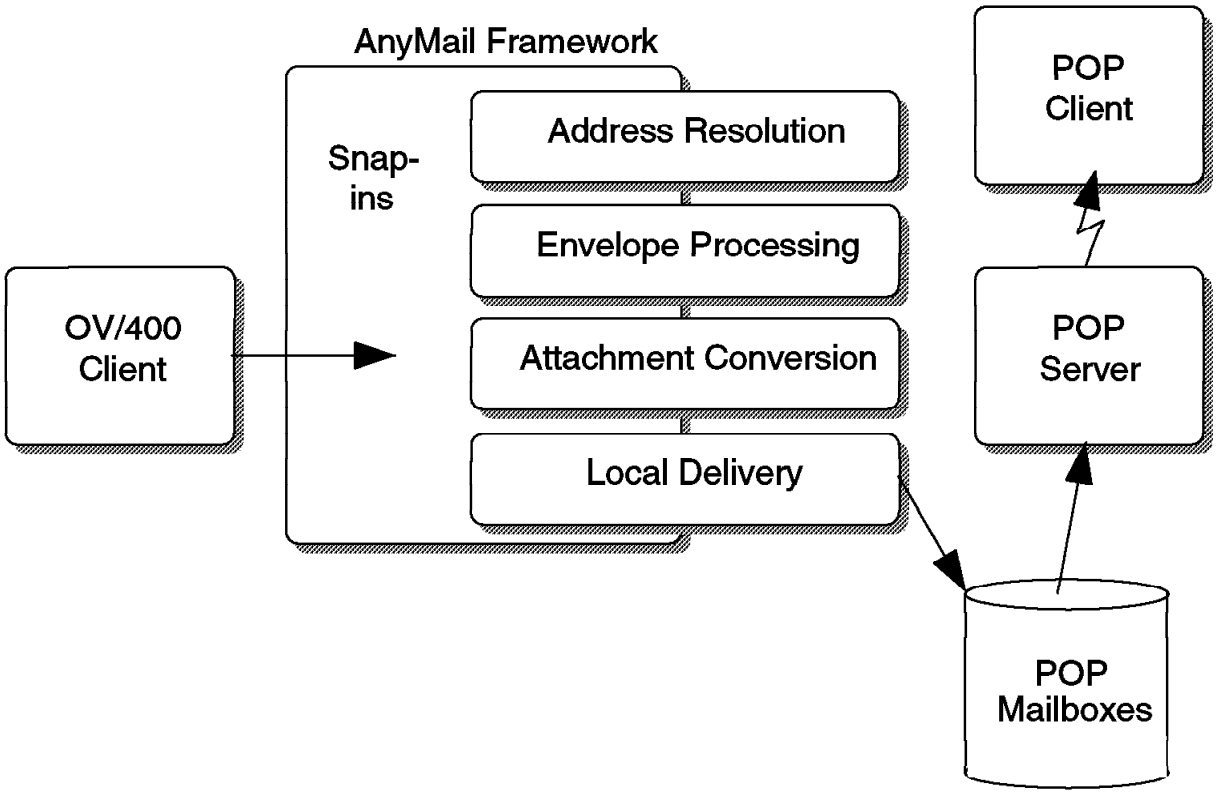


Figure 82. Data Flow - OV/400 Client to Local POP3 Client

We continue our example of the OV/400 user from 4.3.6, “Example of POP3 Client Sending Mail to OV/400 User with GIF” on page 81. This user forwards the GIF file to a POP3 user. Figure 83 shows the **Work with Mail** display that we saw in Figure 78 on page 84.

```
                              Work with Mail

Working with mail for  . . . . . : HALLEEN INTERNUT


 Type options, press Enter.
   2=Revise a copy  4=Delete    5=View    6=Print  8=Work with details
   9=Print options  10=Forward  11=Reply  12=File remote  13=File local
   14=Authority     15=Fill form

                -------From-------                                    Date
OptStatus    User ID Address Description                     Received
    OPENED    HALLEENPINTERNUTNote and GIF file to OV/400 use05/28/96
 10 NEW       HALLEENPINTERNUT(c-960414.gif)                  05/28/96

                                                                      Bottom
 F3=Exit  F5=Refresh  F6=Work with outgoing mail status
 F10=Display new mail  F11=Change user  F12=Cancel  F24=More keys
```

Figure 83. OV/400 Work with Mail (Option 10 to Forward)

We chose **Option 10** to forward the GIF file and specified the SNADS User ID and Address of OS/2 POP3 user as shown in Figure 84 on page 88. We then pressed **F10** to send the mail.

```
                                Forward Mail

Mail description . . . . . : (c-960414.gif)

 Type distribution list and/or addressees, press F10 to send.
  Distribution list  . . . . .                   F4 for list

-----Addressees------
User ID    Address    Description
 OS2USER     INTERNUT


                                                                    More...
 F3=Exit     F4=Prompt  F9=Attach memo slip  F10=Send  F11=Change details
 F12=Cancel  F13=Change send instructions     F24=More keys
```

Figure 84. Forward Mail

Now we go back to OS/2 UltiMail Lite.

- Since our In-basket was open, we clicked on **In-basket** in the title bar to get the pull-down menu.
- We clicked on **Refresh** to refresh my mail.
- Figure 85 shows this display:

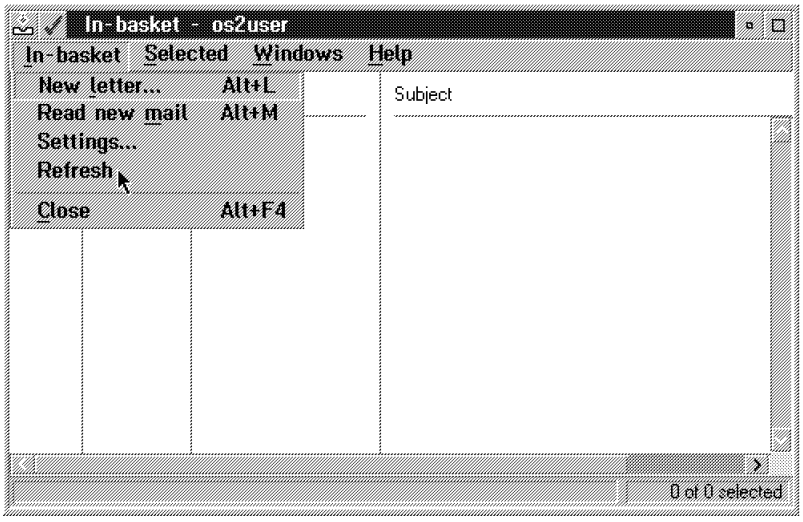


Figure 85. Refresh Pull-Down Menu

The UltiMail Lite POP3 client then made a POP3 server connection to our AS/400 system to retrieve the mail.

Figure 86 on page 89 shows the new mail that was retrieved.

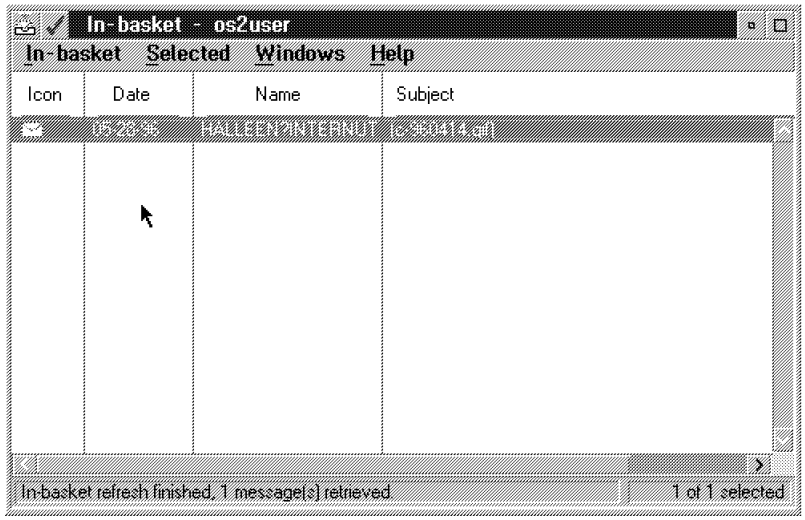


Figure 86. New Mail in In-Basket

We then double-clicked on the new mail item to open the mail. Figure 87 shows the opened mail and shows the GIF file.

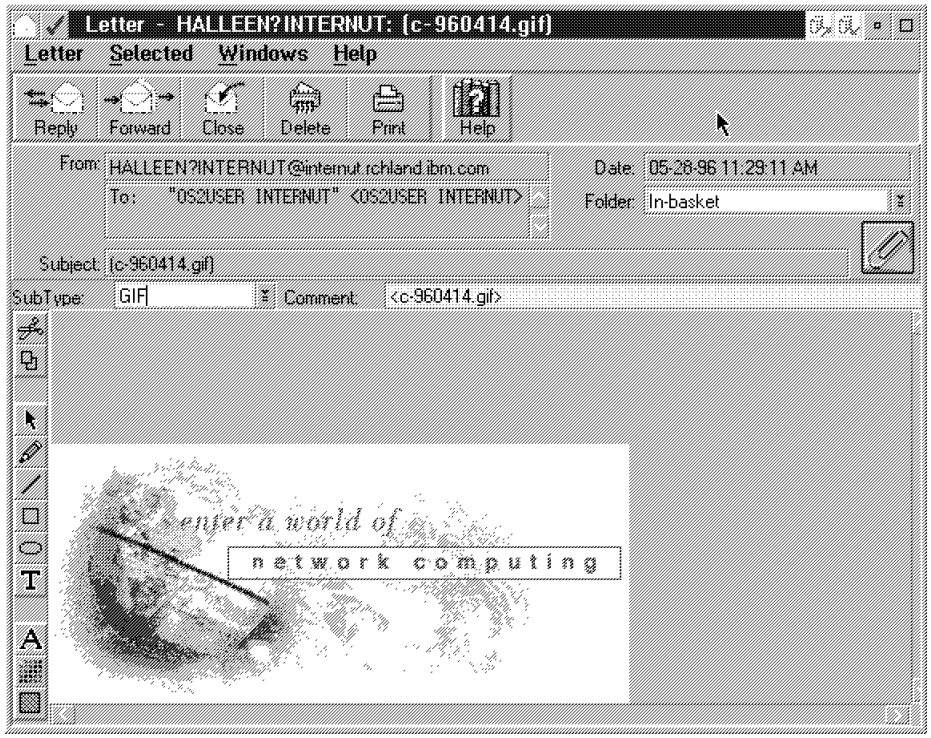


Figure 87. Opened Mail Showing GIF File

Notice that the GIF file has stayed intact since it was sent from Netscape. (Reference Figure 77 on page 84 for a comparison.)

4.4 No TCP/IP, No E-Mail? Not So!

If your network does not include TCP/IP, another alternative may be to connect to Advantis or IGN (IBM Global Network). Some customers are connected directly to Advantis or IGN through an SNA SDLC line to one of the AS/400 communication lines. Similar to Figure 60 on page 67, Advantis can take the role of the SMTP to SNADS gateway and forward mail to your AS/400 system, thus giving you access to Internet mail. The AS/400 system can also be set up to dial Advantis at predetermined times to download mail and reduce line costs. For more information on Advantis or IGN offerings, see their WWW page at

`http://www.ibm.com/globalnetwork`

Chapter 5. Telnet on the AS/400 System

Telnet is the protocol for accessing and using the resources of a remote system on a TCP/IP network as if your workstation (the client) was locally connected to the remote system (the server). The AS/400 TCP/IP supports the Telnet client applications as well as the server applications.

In this chapter, we cover the following three topics:

- AS/400 as a Telnet Client
- AS/400 as a Telnet Server
- Telnet Security considerations

These topics are related to the usage of the Internet and are not step-by-step installation and configuration guidelines for TCP/IP Telnet applications.

Read This!

How to install and configure the Telnet applications on the AS/400 system is already covered in the TCP/IP manual, *TCP/IP Configuration and Reference, Version 3*, SC41-3420.

5.1 The Telnet Client on the AS/400 System

The AS/400 Telnet client application and Telnet server application on the remote system negotiate the transmission of data streams in the following modes and in the order shown:

- TN5250 - 5250 full-screen mode
- TN3270 - 3270 full-screen mode
- VT220 - full-screen mode
- VT100 - full-screen mode

The AS/400 Telnet client supports only full-screen modes so the ASCII line mode cannot be negotiated. If negotiations fail for 5250, 3270, or VT220, the VT100 full-screen mode is selected as the default.

Read This!

The functions given to you depend on the terminal type that is negotiated. When using a display station during a Telnet session, you should be aware of keyboard and display differences. Depending on the mode that has been negotiated, you have to consider the use of keyboard language, character sets, keyboard map, and so on. See *TCP/IP Configuration and Reference, Version 3*, SC41-3420.

Figure 88 on page 92 shows the type of modes that are negotiated between the AS/400 Telnet client and Telnet server applications on different kinds of systems.

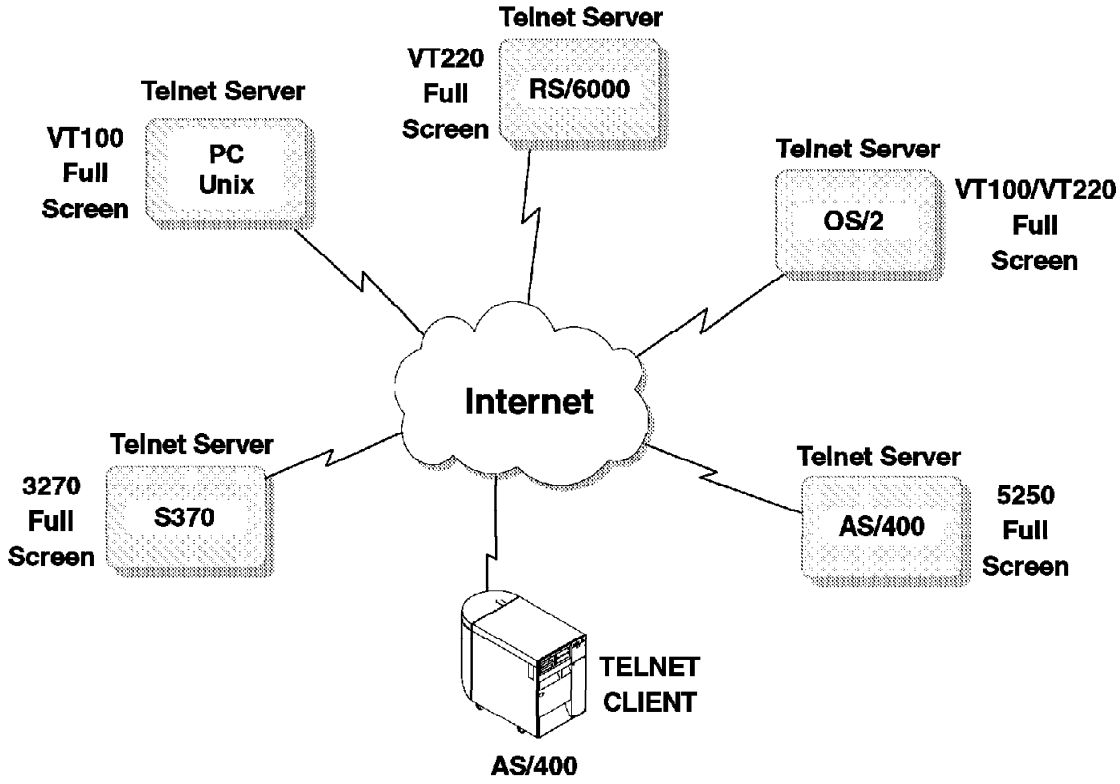


Figure 88. Type of Modes That Are Negotiated for the AS/400 Telnet Client

The Telnet client on the AS/400 system is started by the Start TCP/IP Telnet (STRTCPTELN) command or the TELNET command. Make sure TCP/IP is started. The Start TCP/IP (STRTCP) command activates TCP/IP.

5.1.1 Connecting to a Telnet Server

Let's try to use the AS/400 Telnet client to find some information on the Internet. For example, we can log in to an Internet Service Provider (ISP) to see what they offer and afterwards try to connect to a server that provides information about where to find FTP sites on the Internet.

We start with the Internet Service Provider to find out what they can provide. Execute the command shown in Figure 89 on page 93. It connects you to Real/Time Communications. Real/Time is a service provider, and the server contains information about what they offer.

It is expected that your AS/400 system is already connected to the Internet. See Chapter 2, "Your AS/400 System and the Internet" on page 7 and Chapter 3, "Connecting to Your ISP" on page 23.

```
MAIN                              AS/400 Main Menu
                                                        System:   SYSNM999

Select one of the following:

     1. User tasks
     2. Office tasks
     3. General system tasks
     4. Files, libraries, and folders
     5. Programming
     6. Communications
     7. Define or change the system
     8. Problem handling
     9. Display a menu
    10. Information Assistant options
    11. Client Access tasks

    90. Sign off

Selection or command
===> telnet '198.3.118.42'

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F23=Set initial menu
```

Figure 89. The Telnet Command to Access IP Address 198.3.118.42

Every computer on the Internet may have two addresses: a worded name or a numbered address. The worded name is the host name. The numbered address is the hosts IP address. Every host on the Internet needs an IP address whereas the host name is optional.

Adding the right entry in the host name file or defining a remote name server allows you to use the Telnet command on the AS/400 system as follows:

```
telnet access.realtime.net
```

When the Telnet command is executed and the session is established, you should see the following display:

```
Real/Time Communications (urchin.bga.com) (ttyp0)

login:
```

Figure 90. The Login Display for Real/Time Communications

The display is prompting you for an authorized USERID and a password. Usually both the USERID and the password are required, however some servers are set up to allow anyone to access them.

Use the USERID NEW to log on to the server and press Enter.

```
Real/Time Communications (urchin.bga.com) (ttyp0)

login: new
```

Figure 91. Login USERID for Real/Time Communications

The next display is shown:.

```
Real/Time Communications (urchin.bga.com) (ttyp0)

login: new
Connecting to password server...

Welcome to Real/Time
====================

You have reached our new user registry.  If you wish to open a personal
account, please continue with the registration process.  If you wish to
create an account for your business, please hang up now and call our office
at 451-0046 during business hours.

Any key ...
```

Figure 92. The Welcome Display for Real/Time Communications

You can read what Real/Time is offering. You can page down by using the Enter key.

Try to press the Attention key to see what type of terminal has been negotiated. The following display is shown:

```
                          Send Telnet Control Functions
                                                          System:   SYSNM001

 Select one of the following:

      1.  Interrupt process - IP
      2.  Query connection status - AYT
      3.  Discard host output data - AO
      4.  Clear the data path - SYNCH

      8.  Change VT220 (Primary) keyboard map
      9.  Change VT220 (Alternate) keyboard map

     99.  End Telnet session - QUIT

 ===>


_

  F3=Exit    F12=Cancel
 Primary keyboard map active.
```

Figure 93. Send Telnet Control Functions Display

You are always able to see the terminal type by pushing the Attention key. The displays are different depending on the operating mode. In Figure 93, for example, we see options 8 and 9 that indicate a VT220 terminal mode has been negotiated with Real/Time. If VT100 had been negotiated, for example, these two options would be 6 and 7, and would mention VT100.

When you want to leave the server, press the Attention key and select option 99 to end the Telnet connection.

A lot of the servers on the Internet are UNIX boxes. Therefore, you often see that the mode negotiated is VT100 or VT220. Because of this, you have to be aware of the operational differences between working on a 5250 terminal type and on an ASCII terminal type.

The 5250 is a block mode terminal. Data typed on a 5250 is accumulated in a buffer and *only* sent to the AS/400 system when an AID (Attention IDentifier) key is pressed. An AID key on a 5250 keyboard is a key that initiates a function with the main AS/400 CPU. Everything else such as individual characters, moving the cursor around, or field advance is not seen by the AS/400 CPU.

The following are the AID keys on a 5250 terminal:

- Attention
- Clear
- Command Function 1 through 24 (CF1 through CF24)
- Enter
- Help
- Print
- Record Backspace Function
- Roll Down (Page Down)
- Roll Up (Page Up)

VT100 and VT220 terminals operate in a character mode. Characters are sent *immediately* to the host when a key is pressed.

Let's mention some of the keyboard issues. Typing a control character on an AS/400 keyboard is different than typing a control character on an actual VT100 or VT220 terminal. On a VT100 or VT220 terminal, the control key is pressed and held down while the character associated with the control function is pressed. For example, the VT100 and VT220 Control-C function is entered by pressing the following key sequence:

<CTRL> <C>

In Figure 94 on page 96, this is the lower ASCII terminal type using the character mode. If this user typed Ctrl-C, the Telnet Server sees the Ctrl-C immediately and does something about it, such as terminating the current program or function running in the host system.

Again, look to Figure 94 on page 96 for the 5250 terminal type connected directly to the AS/400 system (does not matter how) through the workstation controller. First, as a 5250 terminal user, you may not even have a Ctrl (Control) key. You might if you are using a PC as a 5250 emulator, but your Ctrl key really does not do anything as the 5250 emulator simply ignores it since it is not part of the 5250 data stream. Second, even if you were to type in some characters, the AS/400 CPU is not notified until an AID key is pressed.

So when using the AS/400 Telnet client support, the equivalent of Ctrl-C is achieved by typing a two-character control indicator followed by pressing the function key associated with *SENDWOCR (Send without Carriage Return) function (the default SENDWOCR is F11).

What the default keyboard map is set to can be checked from the Send Telnet Control Functions display (see Figure 93 on page 94).

The character used to indicate a control character can be selected on the CTLCHAR parameter of the STRTCPTELN command or the Telnet command. The default is the ampersand (&). The &C characters must be the last character typed before pressing the *SENDWOCR function key or the &C is not interpreted as a control character. A control character is only sent when the *SENDWOCR function key is pressed.

So the equivalent to the two key sequence of Ctrl-C (pressed together) of VT220 and VT100 is &C<F11> (pressed in sequence).

An example of the Ctrl-C command is when you use the Telnet client to connect to an RS/6000 system, VT220 emulation is typically negotiated. The Ctrl-C sequence is an important one in AIX to end long running commands, such as PING. It is, therefore, important that you know how to do this before using any RS/6000 commands or other UNIX system commands.

Note!

The &C<F11> keys have to be entered quickly, and it may take several attempts before RS/6000 tasks or other UNIX system tasks accept the input.

More information of the different mode considerations is found in *AS/400 TCP/IP Configuration and Reference*, SC41-3420.

Another difference is the way the data arrives on the display. Data is written to a VT100 or VT220 terminal one character at a time, and you see the data arrive as streams of characters. With the 5250, data is written in blocks, and all or part of the display changes at once.

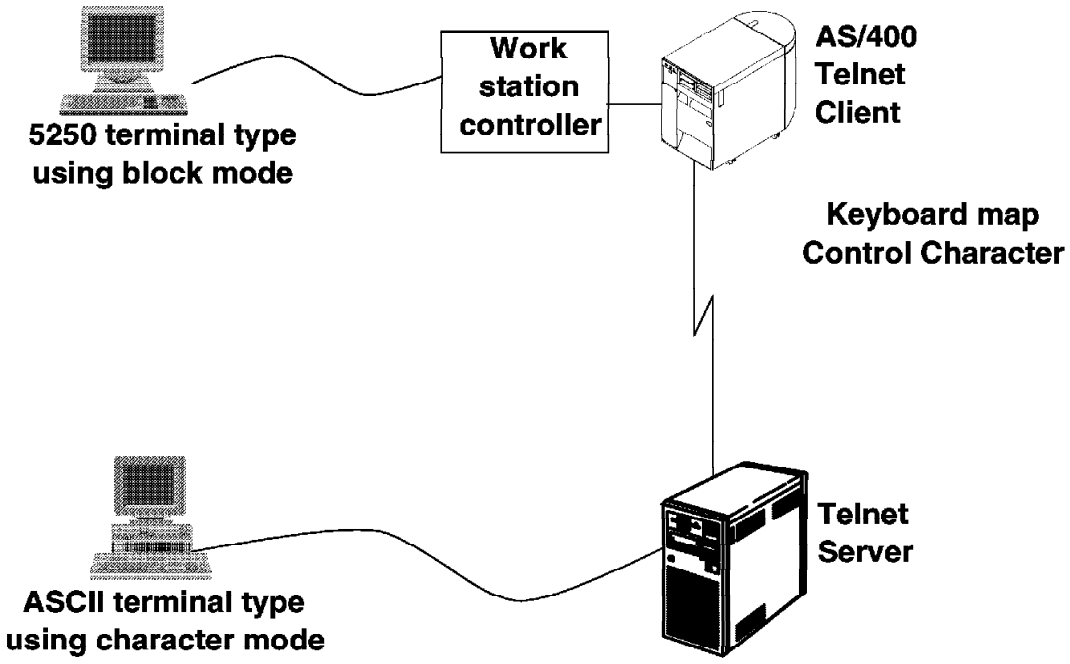


Figure 94. Working as an ASCII Terminal from a 5250 Terminal Type

The other example we wanted to show is to log in to an Archie server to look for FTP sites on the Internet.

Try to do what the following displays in Figure 95 show:

```
MAIN                          AS/400 Main Menu
                                                        System:   SYSNM999
Select one of the following:

     1. User tasks
     2. Office tasks
     3. General system tasks
     4. Files, libraries, and folders
     5. Programming
     6. Communications
     7. Define or change the system
     8. Problem handling
     9. Display a menu
    10. Information Assistant options
    11. Client Access tasks

    90. Sign off

Selection or command
===> telnet archie.unl.edu
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel   F13=Information Assistant
F23=Set initial menu
```

Figure 95. Telnet to Archie.unl.edu

Log in with USERID ARCHIE.

```
SunOS UNIX (crcnis2)

login: archie
```

Figure 96. Login Display for the Archie Server

The password is archie. If you did not want the password to show on the display, press the F6 key prior to typing the password. The F6 key tells the AS/400 system to hide the password on the display. Please remember that even though the password is hidden from view on your screen it will still flow un-encrypted (in the clear) across the Internet with the possibility that it could be seen.

```
SunOS UNIX (crcnis2)

login: archie
Password archie
```

Figure 97. USERID and Password for the Archie Server

After logging in, you see the following display. Try to write server and press Enter, or just see what you can do on your own.

```
################################################################################

        #     # ####### ######  ######     #     #####  #    #     #
        ##    # #       #     # #     #   # #   #     # #   #     # #
        # #   # #       #     # #     #  #   #  #       #  #     #   #
        #  #  # #####   ######  ######  #     #  #####  ###     #     #
        #   # # #       #     # #   #   #######       # #  #    #######
        #    ## #       #     # #    #  #     # #     # #   #   #     #
        #     # ####### ######  #     # #     #  #####  #    #  #     #

     Welcome to the ARCHIE server at the University of Nebraska - Lincoln

    If you need further instructions, type help at the unl-archie> prompt.

################################################################################

# Bunyip Information Systems, 1993

# Terminal type ibm-3180-2' is unknown to this system.
# erase' character is [?'.
# search' (type string) has the value sub'.
unl-archie> server
```

Figure 98. Welcome Display on the Archie Server

When you want to leave the server, you just press the Attention key and choose option 99 to quit Telnet.

```
                          Send Telnet Control Functions
                                                                System:   SYSNM999
 Select one of the following:

      1.  Interrupt process - IP
      2.  Query connection status - AYT
      3.  Discard host output data - AO
      4.  Clear the data path - SYNCH

      6.  Change VT100 (Primary) keyboard map
      7.  Change VT100 (Alternate) keyboard map

     99.  End Telnet session - QUIT

 ===> 99
  F3=Exit    F12=Cancel
 Primary keyboard map active.
```

Figure 99. Send Telnet Control Functions Display

You are now back on your AS/400 system.

5.2 The Telnet Server on the AS/400 System

The AS/400 Telnet server support negotiates the transmission of data with the remote Telnet client application. The following operating modes are supported in the order shown:

- 5250 full-screen mode
- 3270 full-screen mode
- VT220 full-screen mode
- VT100 full-screen mode
- ASCII line-mode

The functions available to you depend on the terminal type that is negotiated. Be aware of the following commands after you log on to an AS/400 Telnet server from a client other than an AS/400 system:

- DSPVTMAP Display VT keyboard map: This command displays the current values for the keyboard mapping of your TELNET VT220 or VT100 session. This can be important when looking for the *ATTN* key or the *SYS REQUEST* key.
- CHGVTMAP Change VT keyboard map: This command changes the keyboard mapping values to be used during a VT220 or VT100 TELNET server session.

Be aware of the operating restrictions, especially for the OfficeVision/400 editor and the ASCII line mode. The restrictions differ depending on the terminal type that is being used.

Every time a Telnet session is established to the AS/400 system, a virtual device is associated with the session. It is used to form a connection between a user and a physical workstation attached to a remote system.

The server support attempts to match a virtual device description with a device type and model similar to the device on your local system. The server automatically selects (and creates it, if necessary) a virtual device description.

The Telnet server support is easy to configure. The Configure TCP/IP Telnet (CFGTCPTELN) command displays a menu that allows you to do the configuration activities. The display is shown in Figure 100.

```
                          Configure TCP/IP Telnet
                                                          System:   SYSNM001

 Select one of the following:

      1. Change Telnet attributes

   Associated system values:
     10. Work with autoconfigure virtual devices
     11. Work with limit security officer device access

 Selection or command
 ===> _

_

 F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
 (C) COPYRIGHT IBM CORP. 1987, 1994.
```

Figure 100. Configure TCP/IP Telnet Display

The options shown are related to a 5250 full-screen mode session. The type of mode session that you are running determines which options you see on the menu. The three menu options are the basic considerations you need to do before starting the AS/400 Telnet server.

After you have configured the Telnet server application, you can start the server.

Type the STRTCPSVR SERVER(*Telnet) command and press Enter.

A server job, QTGTELNETS, is started in the QSYSWRK subsystem. Type the GO TCPADM command and select option 20, Work with TCP/IP jobs in QSYSWRK subsystem, and you see all of the started TCP/IP server jobs.

If you display the job log for the QTGTelnet server job, you see a display similar to the one in Figure 101.

```
                              Display Job Log
                                                          System:   SYSNM001

Job . . :   QTGTelnetS    User . . :   QTCP          Number . . . :   174539


    Job 174539/QTCP/QTGTelnetS started on 11/25/95 at 15:46:09 in subsystem
      QSYSWRK in QSYS. Job entered system on 11/25/95 at 15:46:09.
    Job 174539/QTCP/QTGTelnetS submitted.
    Vary off completed for device QPADEV0002.
    Description for device QPADEV0002 changed.
    Vary on completed for device QPADEV0002.
    Description for device QPADEV0001 created.
    Vary on completed for device QPADEV0001.
    Vary off completed for device QPADEV0001.
    Data area QPADEV0001 created in library QTEMP.
    Object QPADEV0001 in QSYS type *DEVD deleted.
    Description for device QPADEV0001 created.
    Vary on completed for device QPADEV0001.
    Object QPADEV0001 in QTEMP type *DTAARA deleted.
                                                                      Bottom

Press Enter to continue.

F3=Exit   F5=Refresh   F10=Display detailed messages   F12=Cancel
F16=Job menu           F24=More keys
```

Figure 101. Display Joblog for the AS/400 Telnet Job in QSYSWRK Subsystem

It is the server job, QTGTELNETS, that controls the incoming Telnet sessions and activates and deactivates the associated virtual device descriptions. If necessary, it automatically creates new virtual device descriptions depending on the system value for autoconfiguration of virtual devices.

A Telnet user is treated just the same as any other AS/400 user. Users have the authority given to their user profile and the application they are running.

Connecting the AS/400 system as a Telnet server to the Internet is not the most used way of using and accessing the Internet. It certainly has some security impacts. Implementation of a World Wide Web server is the popular way of allowing remote users access to your system for many reasons that are covered later in this book. Security, covered in the next section, is just one of those reasons.

5.3 Telnet Security

When connecting your system to the Internet, security becomes a very important issue. See Chapter 10, “Security and Audit on the Internet” on page 267.

IMPORTANT!

Telnet servers on the Internet are the most exposed systems to crackers (hackers who specialize in gaining access to computer systems). If you do not require any Telnet server jobs starting automatically when you start TCP/IP, type CHGTELNA AUTOSTART(*NO).

The AS/400 Telnet client application is not the critical part when considering the security issue. It is the server application. Running the Telnet server on your AS/400 system opens the door for accessing your system with a login session so you have to be aware of the security aspect. We recommend a security level of 40 when operating the AS/400 system as an Internet server.

What are the security aspects for running your AS/400 system as a Telnet server on the Internet? There are many. Here are some of them:

- Exposure of your AS/400 system
- Exposure of your local intra-network. This includes SNA also!
- Object security
- User profiles and authorities
- Important system values

5.3.1 Exposure of Your AS/400 System

You have to be aware that when you run TCP/IP and the Telnet server, USERIDs and passwords are sent across the Internet. The USERID and password are not encrypted. If anyone is tapping or tracing your communications, they can get these USERIDs and passwords. We recommend that you do not use USERIDs with high authorization across TCP/IP and the Internet.

Be aware of the following system values:

- QAUTOVRT - Autoconfigure virtual devices.
- QMAXSIGN - Maximum sign-on attempts allowed.
- QLMTSECOFR - Limits the devices the security officer can sign on to.

Multiplied with each other, the first two values give the total number of attempts to break into a system before the user is denied access. For example, if QAUTOVRT is set to 50 and QMAXSIGN to 3, you can try to log on to the system 150 times before the virtual device descriptions are disabled and no further access is allowed. Also keep in mind with QLMTSECOFR set to 1, you can limit a user with *ALLOBJ or *SERVICE authority to a specific workstation and deny any remote users with this authority to sign on unless you grant object authority to the security officer for that virtual device (for example: GRTOBJAUT OBJ(display_name) OBJTYPE(*DEVD) AUT(*CHANGE) USER(QSECOFR)).

5.3.2 Exposure of Your Local Network

Allowing users to connect to your AS/400 system can have an impact on the security for any local area network connected to the AS/400 system. Make sure that every possibility to get access to hosts on the LAN is denied.

Check that:

- IP DATAGRAM FORWARDING is not allowed if possible. Type CHGTCPA IPDTGFWD(*NO).

- Exclude Telnet users from having authority to commands that pass through or give access to other hosts on the LAN. Commands such as:
  - STRTCPTELN and Telnet (The Telnet Client)
  - STRTCPFTP and FTP (The FTP Client)
  - STRPASTHR (Start Pass Through)
  - SNDNETF (Send Net File)
  - And so on

5.3.3 Object Security

Everything on the AS/400 system is an object (programs, files, libraries, commands, and so on). Each of these objects can own its own authority. Make sure that users do not have access to objects that they are not allowed to see or use. Learn how the security check is done on the AS/400 system and implement what is necessary by using user groups, authorization lists, public authority (*EXCLUDE), and so on. See *AS/400 Basic Security Guide*, SC41-3301, and *AS/400 Security - Reference*, SC41-3302, for more information.

5.3.4 User Profiles and Environments

The USERIDs that are going to be used for incoming Telnet clients must be created with as little capability as possible. Consider the use of user profile and job description parameters:

- User class
- Initial program
- Initial menu
- Limit capability
- Special authority
- Maximum allowed storage
- Group profile and authority
- Owner
- Home directory
- Initial library list
- And so on

Remember to change the password on every user profile delivered with the system by IBM. Insist that your IBM Customer Engineer does not leave the system with the QSRV password of QSRV, IBMSRV, or SERVICE, for example.

5.3.5 Important System Values

To protect your system against unauthorized users trying to get access to your system through the Internet, the AS/400 system offers a lot of system values that can help you prevent unauthorized users to easily gain access on your AS/400 system and maybe threaten your data, programs, or privacy of your business. They help prevent the unauthorized users from breaking into your system.

The following list contains system values that are important for security when running the AS/400 system as a Telnet server:

- **Sign-on attempts**

System Value	Description
QMAXSGNACN	Action to take for failed sign on attempts.
QMAXSIGN	Maximum sign-on attempts allowed.
QDSPSGNINF	Sign-on display information control.

- **Password Control**

System Value	Description
QPWDEXPITV	Password expiration interval.
QPWDLMTAJC	Limit adjacent digits in password.
QPWDLMTCHR	Limit characters in password.
QPWDLMTREP	Limit repeating characters in password.
QPWDMAXLEN	Maximum password length.
QPWDMINLEN	Minimum password length.
QPWDPOSDIF	Limit password character positions.
QPWDRQDDGT	Require digit in password.
QPWDRQDDIF	Duplicate password control.
QPWDVLDPGM	Password validation program.

- **Creation and Access to device descriptions**

System Value	Description
QAUTOVRT	Autoconfigure virtual devices.
QLMTSECOFR	Limit security officer device access.
QLMTDEVSSN	Limit device sessions.

See the *AS/400 Basic Security Guide*, SC41-3301, and *AS/400 Security - Reference*, SC41-3302, if you want to know more about security and system values.

Chapter 6. FTP on the AS/400 System

This chapter covers the File Transfer Protocol (FTP) client and server. It introduces FTP, and then goes into the FTP server and FTP client in separate sections. This chapter finishes with some good practical examples of using FTP on the AS/400.

Important note

If you would like to follow along with the examples in this and other chapters you first will want to install the ITSO Company demonstration and other web based applications from the CD-ROM that came with this redbook. Please see Appendix A, “Installing the ITSO Company Demo” on page 285 for instructions on how to get your AS/400 up and running right away.

6.1 Introduction to FTP

The File Transfer Protocol (FTP) enables sending and receiving files across a TCP/IP network. Compared to many things in the Internet, FTP is definitely not cool, but it is a good representative of the business side of the TCP/IP applications. FTP was originally designed in the early 1970s as a simple method of transferring files between host systems. It evolved into a ″standard″ method of transferring files in the ARPANET and finally in the TCP/IP protocol suite. Today, FTP allows you to access a very large amount of data that is spread on FTP servers all over the world. These FTP servers can all be different computers (for example, Apple, PC, UNIX, AS/400 system, and MVS) with only one thing in common: *they must support the TCP/IP protocol.*

The FTP on the AS/400 system consists of two parts: the *client* and the *server* function. This is shown in Figure 102 on page 106.

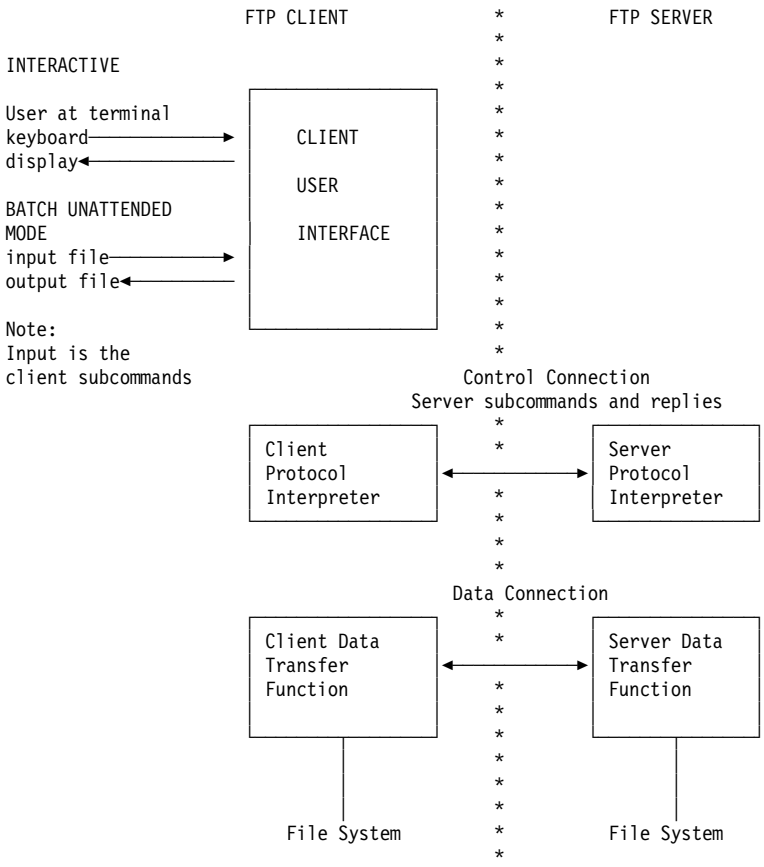


Figure 102. Relationship Between FTP Client and FTP Server

The client initiates FTP *subcommands* that are sent to the FTP server. The results of these subcommands are then displayed. On the AS/400 system, you can enter the subcommands interactively (FTP is designed to be an interactive protocol and interface), or write a simple script for an unattended batch mode operation. In this case, the commands are read from a file and the responses to the subcommands can be written to a file.

To transfer the files between the client and the server, two connections are used. The *control connection* is used to send the subcommands from the client to the server and back again. The second connection is called the *data connection* and is used for transferring lists of file names and the actual file data.

Both the client and the server have a data transfer function that interfaces to the AS/400 file system. These functions read and write to the local file system and pass the file data to the data connection. The AS/400 FTP client and server can use most of the different file systems in the Integrated File System (IFS). This basically opens up the entire system to FTP, except non-file type objects in QSYS.LIB such as programs or user profiles. Please see 6.1.2, “Access to the Integrated File System (IFS)” on page 108 for more information.

Normally to transfer files, you need a user ID on both systems or a special configuration set up by the system administrator. A good way around this restriction is the *anonymous* FTP. It essentially allows anyone in the world to have access to a certain area of disk space in a non-threatening way. Please see 6.1.3, “Anonymous FTP Support” on page 111 for more information.

A typical file transfer works similar to this:

1. The client starts a FTP session.
2. The client requests a file transfer by typing in FTP subcommands.
3. The user interface function of the client reads this requests and passes it to the client protocol interpreter.
4. The client protocol interpreter determines what the client requested and translates this into the appropriate FTP server subcommands.
5. The server protocol interpreter receives the subcommands from the control connection and processes it.
6. The results of each server subcommand are transmitted back to the client in the form of an FTP server reply.

Note that the distinction between FTP client and FTP server is seen from the viewpoint of where the commands are initiated, and not from the viewpoint of where the data resides. Thus, the commands are always initiated from the FTP client session, while the files being transferred may initially reside on either system. That is, the client FTP has the option of two FTP subcommands *PUT* and *GET*. PUT means to send the file to the server, GET retrieves a file from the server and places it on the client host.

When the FTP transfers data, it is being transferred as a data stream, not on a record base. Because the AS/400 system stores a lot of the data in an EBCDIC format while PCs and UNIX systems store it in an ASCII format, the data usually needs to be translated.

There is no support for translating special numerical formats (for example, packed decimal or zoned decimal). Therefore, numbers have to be converted to

alphanumeric characters before you can send or receive the file. Files can also be transferred untranslated as binary data.

6.1.1.1 Code Conversion

The code conversion of EBCDIC to ASCII and back again is normally done automatically by both the AS/400 FTP client and server whenever the transfer file type is ASCII. This is the default setting so you usually do not need to care about this.

When you start either an FTP client (on the FTP command), or when you start the FTP servers (in the FTP attributes), there are three parameters that influence the code conversion process. The first is the CCSID parameter that specifies the ASCII coded character set identifier that is used to translate your AS/400 host′s EBCDIC to and from ASCII. The TBLFTPIN and TBLFTPOUT parameters allow you to specify tables that map either all incoming data or all outgoing data.

The rules for how to use the CCSID, TBLFTPIN, and TBLFTPOUT parameters are defined in the online help text. But, full understanding of what is going on may not be too obvious.

Anytime you let either the FTP server or client on the AS/400 system create the file as part of the transfer to the AS/400 system, FTP may make a wrong assumption when labeling the CCSID of the database file. This causes additional performance penalties as the now local file on your AS/400 system must be (at runtime) converted to your system or job′s default CCSID. So, the rule to follow is *always* create the file on the AS/400 system first. This is the only way you can ensure that the AS/400 system′s FTP support knows how to fill in the data properly.

Another bonus is that when you create the file on the AS/400 system prior to sending the data, the transfer performance is often much better because FTP does not need to buffer the file. It just stores the data directly in the file you created.

6.1.2 Access to the Integrated File System (IFS)

The introduction of the IFS brought us several different file systems on the AS/400 system. These different file systems all have different naming schemes and structures. In the following sections, we discuss the different file systems and give short hints on what they can be used for.

6.1.2.1 Root(/) File System

The root file system supports a directory structure and commands that access information in stream files. It is similar to the QDLS file system but has the ability to use longer file names (up to 255 characters) and has removed some constraints that the QDLS had. For the FTP server, the access to the root file system means mainly to get access to the stream files.

6.1.2.2 QSYS.LIB File System

This file system (normally known as the AS/400 database) allows you to have direct access from FTP clients to the database of the AS/400 system. This means that you can access physical files (PF), logical files (LF), source physical files (SRCPF), and save files (SAVF). Note that you might need to restructure some physical files due to the fact that only alphanumeric fields are allowed with FTP (that is, no packed numeric data).

AS/400 FTP does not allow you to access objects other than the ones listed previously. You cannot, for example, FTP a spooled file or an output queue object. You can, however, save those objects in a save file and then FTP them to the remote system. Hopefully, that remote system is another AS/400 system that knows what to do with a save file!

6.1.2.3 QDLS File System

With QDLS, you have access to ″virtual hard disks″. Here you can store and archive the PC files you may need to distribute to your PCs. But the QDLS is also used with OfficeVision (OV/400). OV/400 is the electronic mailing, archiving, and distribution system on the AS/400 system. So access to QDLS means that you also have access to the mail and the documents you have archived on the AS/400 system.

6.1.2.4 QOPT File System

With the support of the QOPT file system, you have access to the optical disks that you can attach to the AS/400 system. Here we support the CD-ROMs and CD-WORMs.

The CDs are used more and more as a cheap medium to archive huge amounts of data.

6.1.2.5 QOpenSys File System

The QOpenSys file system opens up the AS/400 system for the UNIX world, because this is the file system that is usually used on UNIX machines. This allows us to store files in a native UNIX-like format on the AS/400 system. Note that this allows the AS/400 system to act as an FTP server for the UNIX world. You do not need to convert the files from one format to the other only to allow UNIX clients to transfer files to and from an FTP server.

6.1.2.6 QLANSrv File System

The QLANSrv file system was introduced to allow the Integrated PC Server (formerly known as File Server Input Output Processor (FSIOP)), which is a ″built-in PC″ within the AS/400 system, a fast access to AS/400 disks. This gives you the fastest access available on the AS/400 system to PC-like files. It is in many ways similar to the QDLS file system, but it is many times faster when you access QLANSrv through the Integrated PC Server.

Note: You need additional software (the LAN Server/400) to support this file system.

6.1.2.7 Naming Formats on AS/400 System

File names must be specified in particular formats. These formats vary depending on the file system in which the file resides. In this book, we cannot possibly list all of the different file formats that you may run across, but we can make sure that you understand the AS/400 system′s format.

Two naming formats are supported by the AS/400 FTP. They can be categorized as:

Format	Format Description
NAMEFMT 0	This is the *traditional* (before we knew better) way of naming the physical files, logical files, and source file members that FTP addresses on the AS/400 system. This is a naming format only for the *QSYS.LIB file system* database files. The syntax is `libname/filename.membername` that you can read as library name slash file name dot (or period) member name. This naming format worked fine for the kinds of database objects that FTP originally accessed, but does not work for some of the new objects that we find in the QDLS or QOPT file systems. Since this name format was first, it is called NAMEFMT 0, and is the default whenever you use either the FTP client or server on the AS/400 system. **Note:** If you have PTF SF31879 installed, then the default naming format can change automatically. Please see the labeled box after the description of the naming format 1.

NAMEFMT 1	This is the naming format for the *IFS*. This format must be used to work with the IFS file systems such as QDLS or QOPT and save files found in the traditional file system QSYS.LIB.
	
	This naming format was created when AS/400 FTP was improved to access (in addition to the database file system) QDLS and QOPT. You also can now FTP any AS/400 object that can find its way into a save file. Save files, to be clear, can be sent or received using either the old NAMEFMT 0 or the new NAMEFMT 1.
	
	A path name (also called a pathname on some systems) tells FTP how to locate an object. The path name is expressed as a sequence of directory names followed by the name of the object. Individual directories and the object name are separated by a slash (/) character. Here is an example: `directory1/directory2/file` that is read as directory one slash directory two slash file. Some more real-to-life examples sometimes can help:
	
	`/QDLS/KRISP/MONDO/BIGFILE.TXT`
	`/QSYS.LIB/ITSOIC400.LIB/QRPGLESRC.FILE/FTPLOGON.MBR`
	`/QSYS.LIB/SAVLIB.LIB/SAVEJUN.SAVF`
	`/QOpenSys/Direct/file12` **1**
	`/newclass/l05d040.htm`
	
	1 The QOpenSys path names are case sensitive. Other path names can usually be written as upper, lower, or mixed case.

Note: Because NAMEFMT 1 does everything that the old NAMEFMT 0 does plus a whole lot more, we strongly advise that you only use the new IFS naming format. Unfortunately, the old naming format is the default, so one of the first things you should do when using the AS/400 FTP client is issue the FTP subcommand NAMEFMT 1. And conversely, when you are using the AS/400 FTP server, issue the SITE NAMEFMT 1 (or with some FTP clients that do not support SITE subcommand QUOTE SITE NAMEFMT 1) subcommand from the non-AS/400 client. If you are using FTP between two AS/400 systems, then you need only do the NAMEFMT 1 on the client side as the naming format request is automatically passed to the server by your AS/400 FTP client.

FTP Server Enhanced with PTF SF31879

With PTF SF31879, handling the naming formats in FTP server is enhanced in the following way. If the *first* command sent to the server that requires a file/path name in an FTP session is either blank (no file name provided) or starts with a '/' or a '~', the FTP server *assumes* IFS naming (NAMEFMT 1) for that session. Just in case this happens by accident, the normal reply for the command is "prepended" by an initial reply stating that the NAMEFMT setting has been set to 1 as an aid to warn the client.

6.1.2.8 Library File System for AS/400 Database Files and Save Files

Here are the general naming rules for naming format 1 (the IFS format) on the AS/400 system.

Each part of a database name must be qualified with the appropriate extension as indicated in the following table:

Table 3. Object Type Extensions

Name Part	Object Type Extension
Library	.LIB
File	.FILE (generic extension for file name) .PF (physical file) .LF (logical file) .SRCPF (source physical file) .SAVF (save file)
Member	.MBR

Note: The file name part can be qualified with either the generic .FILE extension or a specific file type extension. An example is .PF for a physical file.

6.1.3 Anonymous FTP Support

As the Internet started to grow to thousands of servers, there was an urgent need to have an easy access to this huge amount of data. Because you cannot administer to the millions of potential users in all parts of the world, you have to find an easy and simple way to allow access to this "public" data. One of these is the *anonymous FTP* method.

Most computer systems in the Internet offer this support. Normally, if you log on to a computer, you are asked for a user ID and a password. But on the Internet, you probably log on to a hundred different computers distributed all over the world. To be useable for your daily work, you now need a common user ID. This common user ID is: anonymous.

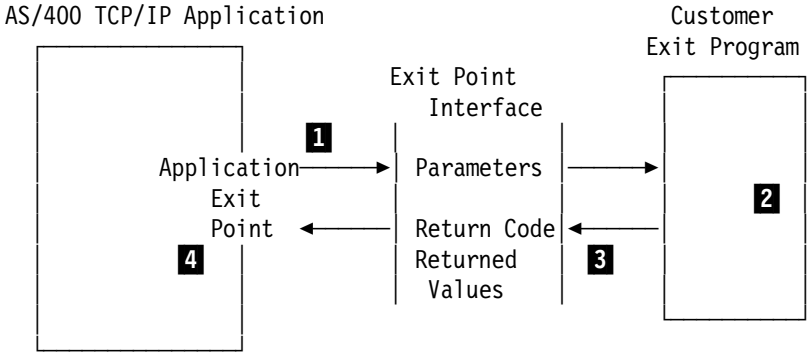
So when the server asks you for a user ID, put in `anonymous`. Custom and etiquette dictate that you use your E-mail address (for example, `myname@ibm.net`) as your password. Some servers may accept any character string such as "`asdfasdf`" as your E-mail address, but do not use that because some servers may insist on a formal correct E-mail address and others may prohibit further access to your IP-address. If the server responds with the message "not a valid password", this usually means that the password is scanned for the '@' character. So you have to respond with a formal valid E-mail address.

The anonymous FTP servers usually contain software, documents, graphic images, or any other kinds of information. All of this information is considered to be public accessible and can be read by everyone who wants to. Note that this may be subject to change. The person who ″owns″ the information and the system can, of course, shut down the machine at any time and refuse further public access. You cannot do anything about it.

On the AS/400 system, the anonymous FTP is enabled by using an FTP server logon exit program and an FTP server request validation exit program. When an anonymous FTP is enabled on the AS/400 system, the FTP server requests an E-mail address instead of a password in an anonymous logon.

6.1.4 FTP Exit Programs

An *exit point* is a specific point in the TCP/IP application where control may be passed to an exit program. The FTP exit programs are used to control the use of the FTP server and FTP client. These exit programs offer additional security and transaction logging on an AS/400 FTP server.



Processing flow:

1 TCP/IP application passes request parameters to the exit program.
2 Exit program processes request parameters
3 Exit program returns information to the TCP/IP application.
4 TCP/IP application performs operation based on exit program response.

Figure 103. TCP/IP Exit Point Processing

Three different exit points are provided for FTP. The first one is used to validate requests processed by the FTP client program, the second one is used to validate requests processed by the FTP server program, and the third one controls the logon requests to the FTP server program.

Table 4 (Page 1 of 2). FTP Exit Points

FTP Application	Exit Point	Exit Point Format
FTP Client	QIBM_QTMF_CLIENT_REQ	VLRQ0100 1
FTP Server	QIBM_QTMF_SERVER_REQ	VLRQ0100 1

Table 4 (Page 2 of 2). FTP Exit Points		
FTP Application	**Exit Point**	**Exit Point Format**
FTP Server	QIBM_QTMF_SVR_LOGON	TCPQ0100
1 The same format is used for both the FTP Client and FTP server for request validation. This enables the use of one program for both client and server request validation.		

Note: To enable an anonymous FTP, you must define exit programs for *both* FTP server exit points.

The FTP server logon exit program permits or denies a logon to an FTP server based on one or more of the following:

- User ID
- Password
- Remote IP address

FTP request validation exit programs (FTP Client or FTP Server) permit or deny a specific FTP operation based on one or more of the following:

- User profile
- Remote IP address
- Directory, library, files (path names)
- CL commands

Exit programs are defined for exit points using the OS/400 registration facility. You can use Work with Registration Information (WRKREGINF) command to display a list of exit points or simply use Add Exit Program (ADDEXITPGM) command.

```
                      Work with Registration Information

Type options, press Enter.
  5=Display exit point   8=Work with exit programs

                          Exit
     Exit                 Point
Opt  Point                Format    Registered  Text
 _   QIBM_QRQ_SQL         RSQL0100     *YES     Original Remote SQL Server
 _   QIBM_QSY_CHG_PROFILE CHGP0100     *YES     Change User Profile
 _   QIBM_QSY_CRT_PROFILE CRTP0100     *YES     Create User Profile
 _   QIBM_QSY_DLT_PROFILE DLTP0100     *YES     Delete User Profile - after d
 _   QIBM_QSY_DLT_PROFILE DLTP0200     *YES     Delete User Profile - before
 _   QIBM_QSY_RST_PROFILE RSTP0100     *YES     Restore User Profile
 _   QIBM_QTF_TRANSFER    TRAN0100     *YES     Original File Transfer Functi
 _   QIBM_QTMF_CLIENT_REQ VLRQ0100     *YES     FTP Client Request Validation
 _   QIBM_QTMF_SERVER_REQ VLRQ0100     *YES     FTP Server Request Validation
 _   QIBM_QTMF_SVR_LOGON  TCPL0100     *YES     FTP Server Logon
 _   QIBM_QTMT_WSG        QAPP0100     *YES     WSG Server Sign-On Validation
                                                                      More...
Command
===>
F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 104. Work with Registration Information

6.1.4.1 FTP Server Logon Exit Program

The FTP server logon exit program allows you to enforce your own rules for handling logons to an FTP server. The exit program receives as parameters:

- Application identifier
- User identifier
- Authentication string (password or E-mail address from FTP logon)
- Remote IP address
- Length fields for some of the previously mentioned parameters

The exit program returns a parameter whether the logon is rejected, accepted, or if the logon operation should be continued. When the logon is accepted, a user profile is returned and a new current library can be returned as a parameter to override the one specified in the user profile. When the logon is continued, the program can return a new user profile, a new password, and a new current library as parameters to override either entered values or the one specified in the user profile.

Note: The password on the output parameter *must* match the password specified in the user profile for the logon to succeed, but we strongly recommend that passwords *never* be coded in an exit program.

Overriding the initial current library of a user (set in the user profile) is called direct routing. The direct routing support allows you, depending on the clients network address, to route the client directly to a specific library immediately following the logon on the AS/400 FTP server. This allows you, for example, to route sales representatives to their own library. In such a private library, you can keep the files that are unique for a specific sales representative.

The exit program can control *the access to the server* based on the requester′s address or user ID. The exit program can also be used to force your anonymous FTP user to give a valid-looking E-mail address as their password (one that contains a ′@′ character). The exit program cannot give any informative messages to the clients, so they see only that a logon is either accepted or rejected.

Specifying a FTP server logon exit program is one-half of enabling the anonymous FTP. You should create a separate user profile for this. We strongly suggest that this user profile has a password of *NONE. In the exit program, you can also force some of your local users to use this user profile for FTP. All server logon requests can and all anonymous server logon requests *should* be logged to see who accesses your AS/400 system. Because the exit program receives a valid user password as a parameter, care should be taken what information to log. The importance of logging is even greater if you are connected to the Internet. You might be compelled later on to maintain a ″black list″ of IP addresses whose owners have tried to do something harmful and block their entrance to your system. Although most of the general Internet users can be considered to be quite harmless, there are the fringe groups that just might have something against your area of business.

Please see 6.6.2.1, “FTP Logon Exit Program (ILE RPG)” on page 136 for an example of a FTP logon exit program that enforces the previously mentioned rules.

6.1.4.2 FTP Client and Server Request Validation Exit Programs

Because both client and server request validation exit programs use the same exit point format (parameters), it is easier to describe their functions in the same chapter. Exit point programs can be one program, but they can also be two separate programs.

The FTP request validation exit program gives you control over whether an operation (that is, an FTP subcommand) is performed or not. The decisions made by the exit programs are in addition to any validation performed by the application program. When installed, the exit program is called each time one of the following requests are processed:

- Session initialization/login (server exit program only during session initialization).
- Directory/library creation.
- Directory/library deletion.
- Setting current directory.
- Listing file names.
- File deletion.
- Sending a file.
- Receiving a file.
- Renaming a file.
- Executing a CL command on the FTP server.

The exit program receives as parameters:

- Application identifier (server or client request)
- Operation identifier
- User profile
- Remote IP address
- Some operation specific data (path name, CL command, and so on)
- Length fields for some of the previously mentioned parameters

It returns as a parameter whether the operation is rejected or accepted. An operation can be rejected completely for the remainder of the session, rejected this time, allowed this time, or allowed unconditionally for the remainder of the session.

Your user exit program can be based upon the user′s IP address and then provide different access to AS/400 data. For example, your employees may have unlimited access but your customers are allowed to see (and ″get″) only the product information that is available to the public. You can also limit FTP users from using certain CL commands, but allow other commands to be used.

The other half of enabling the anonymous FTP is specifying an FTP server exit program. You should create a good protection scheme against your FTP clients. Although the users might be your customers, you must consider them to be threats. Remember that one of the common threats to a computer system is users who does not really know what they are doing but has too much access authority. A good idea is to limit their access to downloading files from selected libraries or directories. The need to upload files must be considered very carefully. If you allow it, limit uploading to selected libraries or directories that are not the same as download directories. Limit the use of CL commands and deleting objects as well as limiting renaming and creating objects. All in all, limit everything that you do not explicitly allow. And remember that a too tight security scheme is always better than a bad and leaking one. You might also

want to use symbolic links in your ″public″ directory and have the files actually reside in a more secure directory or folder.

The question of using a client exit program is quite different from the server program. Does the task of protecting a server belong to the administrators of the server system? That is quite correct, but if you are connected to the Internet, do you want your employees to search all day for data in the vast selection of FTP servers? If you have business needs for some data that can be found in the Internet, then allow only selected users to do the retrieval. Otherwise, you might have to buy more storage to accommodate all of the neat things your users have found in the Internet.

The functions that are executed during a validation of a request are not restricted. So a lot of nice things, but also not so nice things, can be done. As an example, your exit program can save a library to a save file whenever this save file is requested by a client. Another example might be to refresh the contents of a database file immediately prior to the client′s attempt to download it. But although these functions are nice (and perhaps cool) they must be considered to be ″Trojan Horses″. A ″Trojan Horse″ is a function that is built into a program and triggered by some event usually to do something harmful. So care must be taken when building these functions to the exit programs. It is also good to check regularly that the exit programs used are really the right ones.

Please see 6.6.2.2, “FTP Request Validation Program (ILE RPG)” on page 139 for an example of an FTP client and server request validation exit program that enforces the previously mentioned rules and uses a ″benevolent Trojan Horse″ to refresh contents of a save file.

6.2 FTP Server on the AS/400 System

As already explained in chapter 6.1, “Introduction to FTP” on page 105, the AS/400 system can be used as a server and as a client. In this section, we discuss the AS/400 system as an FTP server. The distinction between an FTP server and an FTP client is from the viewpoint of where the commands are initiated, and not from the viewpoint of where the data resides.

When the FTP server is running on an AS/400 system, the FTP subcommands are issued from a remote client. Whether the remote system is another AS/400 system , a PC, or a UNIX computer is of no interest from the standpoint of the AS/400 FTP server.

6.2.1 Why Use an AS/400 FTP Server

Here are some arguments that differentiate the AS/400 server implementation from other servers on the market:

- Support of several different file systems on *one* system (IFS).
- Built in high security standards.
- AS/400 programs and CL commands can be called through an RCMD subcommand.
- Transfer of folders and documents in the QDLS file system.
- Sending or receiving physical files, source files, logical files, and save files (SAVF).

- Creating and deleting libraries, files, and members using the AS/400 FTP server subcommands.
- Creating and deleting folders using the AS/400 FTP server subcommands.
- If both the server and client are AS/400 systems, you can distribute most AS/400 object types as SAVFs.
- High availability of the FTP server through mirroring, RAID-5, and other AS/400 system availability options.

6.2.2 Where to Use an AS/400 FTP Server

Here are several scenarios where the AS/400 system can be used as an FTP server:

- The AS/400 system can be used as the gateway to the Internet. This is of special interest if you have no LAN installed. In this case, the AS/400 system can be used as a client to ″get″ information from the Internet and as a server to distribute this information to the attached PCs (that is, PCs that are attached through an old emulation card).
- A company wants to make their product information available for the rest of the world. All product information is already in AS/400 format available, for example, as documents in QDLS or as text files in a source member. Of additional value is the fact that the data processing people are familiar with the AS/400 system. This can save a lot of money you may otherwise have to spend in education.
- If several AS/400 systems are connected to each other, then FTP might be the most simple and easiest method for transferring files. The advantage of FTP over SNADS is that there is no store-and-forward. For large files, this can significantly save on transmission time and DASD space.

6.2.3 FTP Server Checklist

- Do you have enough DASD space?
- Is your AS/400 system secured against hackers or crackers? Every server running on your AS/400 system increases the chance of system penetration.
- Is your AS/400 system on the Internet separated from your network?
- Does your company have a lot of public available information to distribute?
- Are your customers interested in having access to your FTP server?
- How can you make it attractive to access your FTP server?
- Is it of value for your company to know who contacted you?
- Is your data processing staff educated in TCP/IP, Internet, WWW, and so on?
- How do you load the data on your FTP server?
- Does your customers already have access to the Internet?
- How many users are going to use FTP service?
- Are you on a LAN or should you buy a LAN File Server instead?
- Did your customers already request access through Internet?
- Do you have customers all over the world?
- Do you need to be on line 24 hours a day?
- Do you know what it costs to be on line?

- Can your afford *not* to be on the Internet?

6.2.4 FTP Server Security

If you use your AS/400 system as an FTP server on the Internet, be aware that it is accessible for the entire world. You can be rather sure that hackers will try to attack your computer. This is one reason why you *must* do something to secure your computer. Security is discussed in greater detail in Chapter 10, “Security and Audit on the Internet” on page 267.

We can give you some hints (without going into too much detail) about which points have to be considered. Note that this is not a complete list.

- Is your Internet AS/400 system connected to any other system?
- Did you secure your AS/400 system or your network?
- Do you control the audit files on a regular base?
- Did you change **all** default passwords?
- Think about not forwarding IP datagrams (set by CHGTCPA).
- Control how many FTP servers you want to allow (set by CHGFTPA).
- Set a proper value for INACTTIMO (set by CHGFTPA).
- Have you implemented an object level security scheme?
- How reliable are your employees?

For additional security information, see Appendix C2 in the *OS/400 TCP/IP Configuration and Reference*, SC41-3420.

We advise that you do not use your AS/400 system as an FTP server on the Internet without the following safeguards:

1. A firewall should exist between the AS/400 system and the Internet.
2. Use a non-production AS/400 system to be your FTP server system. This AS/400 system should not be network-attached to the rest of your company′s LANs or WANs.
3. The user exit programs must be coded and tested to ensure that no security holes exist.

Even though your AS/400 system is not be connected to the Internet, implement your own security rules and standards through the use of exit programs. Please see 6.1.4, “FTP Exit Programs” on page 112 for more information.

6.2.5 FTP Server Administration

There is not much work needed to administer the FTP server. The FTP server job (or jobs) must be started in the subsystem QSYSWRK before you can use the function from a client.

Figure 105 on page 119 shows the display that appears when you use the Change FTP Attributes (CHGFTPA) command.

```
                        Change FTP Attributes (CHGFTPA)

Type choices, press Enter.

Autostart servers  . . . . . .   *YES          *YES, *NO, *SAME
Number of initial servers  . . .   3            1-20, *SAME, *DFT
Inactivity timeout . . . . . .   300            0-2147483647, *SAME, *DFT
Coded character set identifier     00819        1-65533, *SAME, *DFT
Server mapping tables:
  Outgoing EBCDIC/ASCII table  .   *CCSID       Name, *SAME, *CCSID, *DFT
    Library  . . . . . .                       Name, *LIBL, *CURLIB

  Incoming ASCII/EBCDIC table  .   *CCSID       Name, *SAME, *CCSID, *DFT
    Library  . . . . . .                       Name, *LIBL, *CURLIB
                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 105. Change FTP Attributes (CHGFTPA) Display

Use *YES for the Autostart server's parameter because then three (the default) FTP servers start whenever TCP/IP is started on the AS/400 system using the STRTCP command.

Three initial servers are usually sufficient. If you know that you are expecting a maximum of 10 clients to access your AS/400 server, then you can increase this number. Jobs in waiting do not usually cause serious problems for the AS/400 system.

The CCSID parameters cause FTP to convert the EBCDIC-based characters to and from ASCII code pages. The default (00819) causes all incoming ASCII characters to be translated to the EBCDIC 500 code page. You can specify your own translate table also. Please see the online help and the manuals for more information.

If no FTP server jobs are running, start them with the STRTCPSVR *FTP command. This uses the attributes that you have configured in the Change FTP Attributes (CHGFTPA) display.

You can end the FTP servers with the ENDTCPSVR *FTP command. You cannot end one individual FTP server with this command, but you can use the ENDJOB command if you want to stop only one of the three FTP servers, for example.

6.3 FTP Client on the AS/400 System

In the previous section we discussed FTP on the AS/400 system as a server. This section, however, discusses what can be done with the very capable FTP client.

6.3.1 Useful AS/400 FTP Client Subcommands

The following list contains most of the useful FTP client subcommands. While you are in an FTP session with the AS/400 system, you can always type HELP or use the HELP subcommand for more details than we have provided in this book.

Subcommand	Subcommand Description
AScii	Switch to ASCII mode. This is the default mode and is used for transferring character text. The FTP client translates the incoming ASCII to EBCDIC and outgoing EBCDIC to ASCII. The remote FTP server thinks it is talking to a native ASCII client.
Binary	Switch to binary mode (or sometimes called image transfer). This is used for transferring binary files such as ZIP files, SAVF, or files with packed decimal data.
CD	Change current directory on the *remote* computer.
LCd	Changes the current directory on your *local* computer.
CDUp	Change to the parent directory on the *remote* computer.
Dir	List the files in the *remote* computer (same as in PC).
LS	Same as DIR but lists only the file names (same as in UNIX).
Get	Copies a file from the *remote* computer to the *local* computer.
MGet	Copies multiple files from the *remote* computer to the *local* computer.
PUt	Copies a file from the *local* computer to the *remote* computer.
MPut	Copies multiple files from the *local* computer to the *remote* computer.
LPWd	Shows the present working directory (pwd) on your *local* computer.
PWd	Shows the present working directory (pwd) on the *remote* computer.
DEBug	To control the display of server subcommands sent to the server. **Note:** Notice all of the highlighted *local* and *remote* words in the text above and below this point. The reason they are highlighted is that it is sometimes hard to learn just what each FTP subcommand sends to the remote system and what it gets back. By using the DEBUG subcommand, you can see the English-based communication traffic between the client and the server. This, for example, can help you remember if CD changes the directory on the remote or local system. Note that you can toggle the debug command on and off. With DEBUG 100, you set an FTP client trace on. This results in trace data being dumped and formatted in a file named QPSRVTRC.
LOCstat	Shows the local status information.
NAmefmt	Changes the naming format from the default of 0 (which is the traditional way of naming the database objects in AS/400 libraries to 1 (which is the new IFS naming format).
NOop	To find out if the remote FTP server is responding.
SYSCMD	To run a local AS/400 command. This is very useful.

SITE To send information to the server that it needs for services. Generally, you do not use this command from the AS/400 client that much.

QUOTE To send a server subcommand directly to the server. It is sometimes useful to issue the command `QUOTE HELP PUT`, for example, to get help information for the PUT subcommand from the remote server.

HELP To get online help from the local AS/400 host.

? To get general help from the local AS/400 host.

6.4 Using WWW Browsers as Clients to AS/400 FTP

PTF SF31879 Required

With the birth of World Wide Web, the web browsers have become the main tool for accessing the resources in the Internet. The FTP client function has also been added to their arsenal. The web browsers allow an anonymous FTP connection as well as a usual user ID and password based on the security verification scheme. While you are using the FTP function of a browser to access an AS/400 FTP server, you must remember that you are actually using the FTP server and not the HTTP server. So all of the security measures are in use including the exit programs. If you have not enabled the anonymous FTP on your AS/400 FTP server, then you must use URLs that contain a user ID.

For FTP access, the URLs have a service ID `ftp://`. The general syntax for an FTP URL is:

```
ftp://user:password@host:port/url-path
```

Of this, only the `ftp://host/url-path` in mandatory. With the URL in its most basic format, you connect to the *host* using an anonymous FTP and are shown the directory of the *url-path* unless it fully defines a file name. If the *url-path* defines a file, then it is either shown to you by the browser or handled in some other way depending on your browsers settings. All in all, the file is handled in the same way as if it had been retrieved using an HTTP client.

When you are using an anonymous FTP, you might have to change your browser setting to ensure that the browser sends an E-mail address to the server at logon time. This is mandatory only if your FTP logon exit program requires a valid E-mail address as a password.

If you want to use a user ID-based access to the FTP server that often gives you more rights to access files, you have to enter a user ID or a user ID and password combination in the URL. We do not recommend that you embed your password in links in your external or internal web pages. If the password is not part of the URL, it is up to the browser to request it from the user.

All in all, this method gives you a flexible way to implement interactive access to data from your web pages. And you do not have to teach your employees or your customers to use any new tools when they are using your pages.

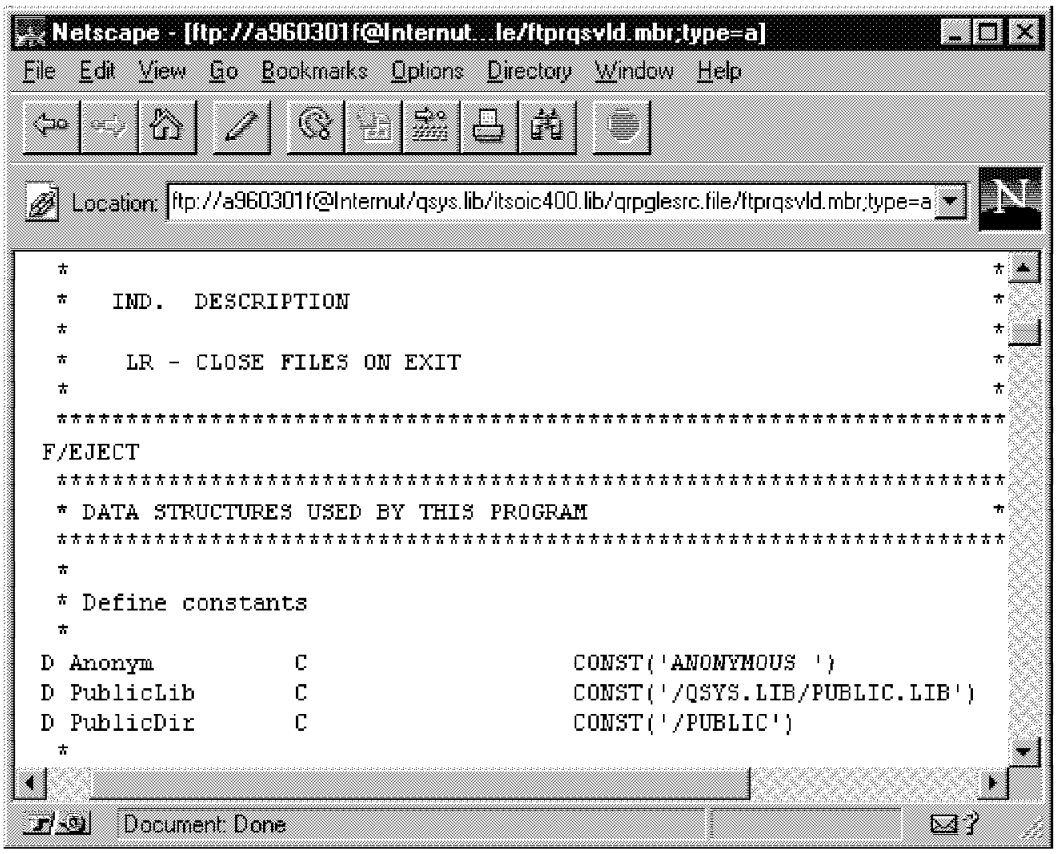


Figure 106. Browsing a File with Web Browser

The URLs for an AS/400 FTP server conform to previously mentioned general standards. When you use them, you have access to all of the data that you have using a usual FTP client. Please note that there are some special considerations using them. The following examples show valid URLs for an AS/400 FTP server:

```
ftp://a960301@internut/Public
```

This URL is interpreted by connecting to the *internut* FTP server, logging in as user *a960301*, prompting for a password, and finally showing the index of a directory called *Public*.

```
ftp://a960301@internut/qsys.lib/itsoic400.lib
```

This URL is interpreted similarly as the previous one and shows an index of all of the files in the AS/400 library *ITSOIC400*

```
ftp://a960301@internut/qsys.lib/itsoic.lib/qrpgsrc.file/ftplog.mbr;type=a
```

This URL shows the member *FTPLOGON* in file *QRPGLESRC* in library *ITSOIC400*.

Note: The `;type=a` at the end of the URL is required to make database files readable. In effect, the web client sends the type ASCII FTP subcommand to the AS/400 server which causes the AS/400 server to convert the EBCDIC data to ASCII.

```
ftp://internut/public
```

This URL is similar to the preceding ones, but there is an anonymous logon to the server. The index of directory *public* is shown if the anonymous user is allowed to access it.

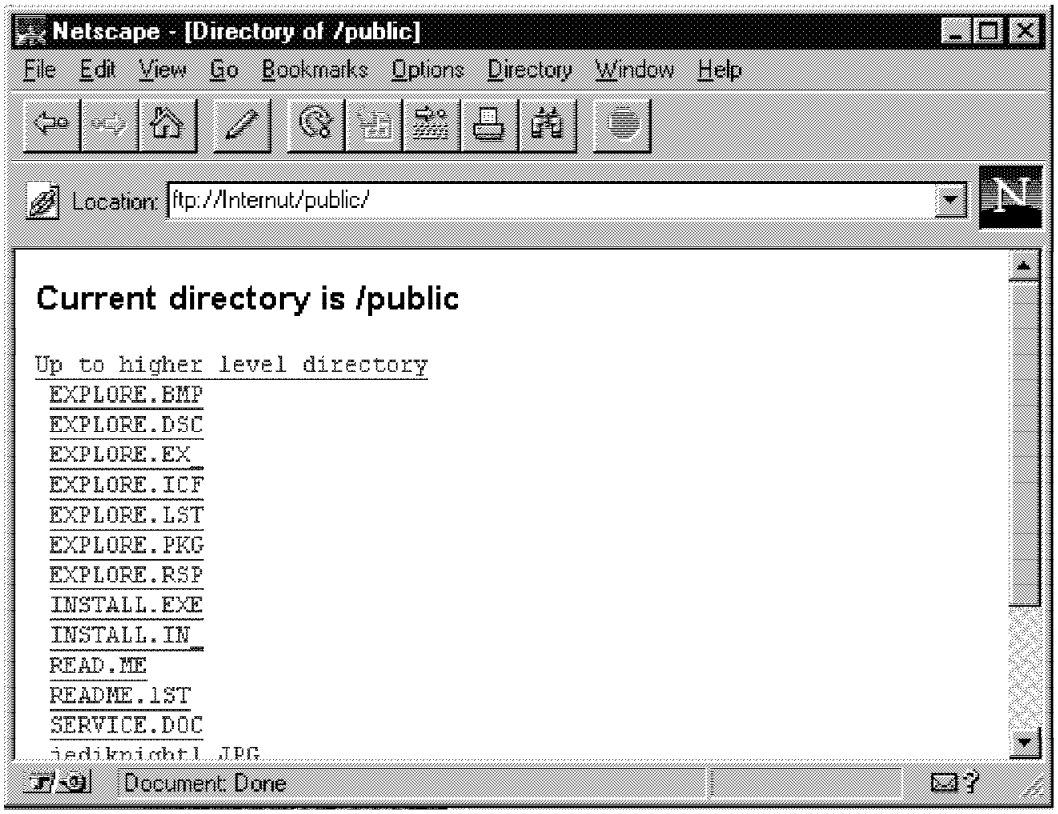


Figure 107. Anonymous FTP with Web Browser

This feature has been tested using Netscape Navigator Gold versions 2.01, 2.02 and 3.0, beta 4, and IBM WebExplorer version 1.1b. They all performed about the same.

Limitations and special considerations:

1. If you want to access the root (′/′) directory, be sure that your first URL in the ″session″ is similar to `ftp://host/directory/`. Otherwise, all subsequent attempts to get to the root do not work correctly. After the first access, you can use the URL without the trailing ′/′.
2. We have also found difficulties with this function when using the host name fully qualified with the domain name. This function simply does not work in this case.
3. PTF SF31879 is needed for this section because of a special change it brings to the AS/400 FTP server. If the PTF is applied, this change happens whether or not the client is a graphic web browser using the ftp:// URL or the old text based FTP client. As you learned in 6.1.2.7, “Naming Formats on AS/400 System” on page 109 the AS/400 has two naming formats. The original (and default) name format 0, or the new name format 1. It is with the new name format 1 (or NAMEFMT 1) that we are allowed to access all the AS/400 file systems including the integrated file system.

   So, from the client′s point of view, they will want to change the AS/400 FTP server to name format 1 as soon as possible to be able to access all the file

systems. The syntax, however, to make this change to the new name format 1 is different on some clients. From an AS/400 the command is `namefmt 1`. From an OS/2 Warp PC you will need to use `site namefmt 1`.

What about from a web browser? Well, it turns out that namefmt is not part of the formal FTP specification. So, as it turns out, there is no way for a graphic web browser (as an FTP client) to automatically issue a `site namefmt 1`. No way, you would think, to have your AS/400 FTP server put or get files in a file system other than QSYS.

With PTF SF31879, handling the naming formats in FTP server is enhanced in the following way. If the *first* command sent to the server that requires a file/path name in an FTP session is either blank (no file name provided) or starts with a '/' or a '~', the FTP server *assumes* IFS naming (NAMEFMT 1) for that session. Just in case this happens by accident, the normal reply for the command is "prepended" by an initial reply stating that the NAMEFMT setting has been set to 1 as an aid to warn the client.

If you have installed the three day class that came as part of the CD-ROM with this book, you could try the sample FTP URLs on this page: `http://yourhost.domain/class/ftp050.htm`.

Note: Some web browsers support uploading of files using FTP. According to my experiments, this function works with the AS/400 FTP server.

End of PTF SF31879 Required

6.5 Using WWW Browsers and AS/400 HTTP Server Directory Access

As I mentioned in 6.4, "Using WWW Browsers as Clients to AS/400 FTP" on page 121, the web browsers can be used with the AS/400 FTP server. Another way to access data in the AS/400 system is to use the AS/400 HTTP server's directory access. This method gives the user similar access to the files as with the web browsers FTP client, but without the limitations caused by exit programs. The scope of this access is defined in the AS/400 HTTP server configuration (please see 8.2.2, "HTTP Server Directives" on page 166 for more information) and by the authorities of user profile QTMHHTTP.

The user is shown an index of the files in the directory, an index of members in a database file, or the contents of a database file member. Access to the QSYS.LIB file system is limited. An index of a library cannot be shown.

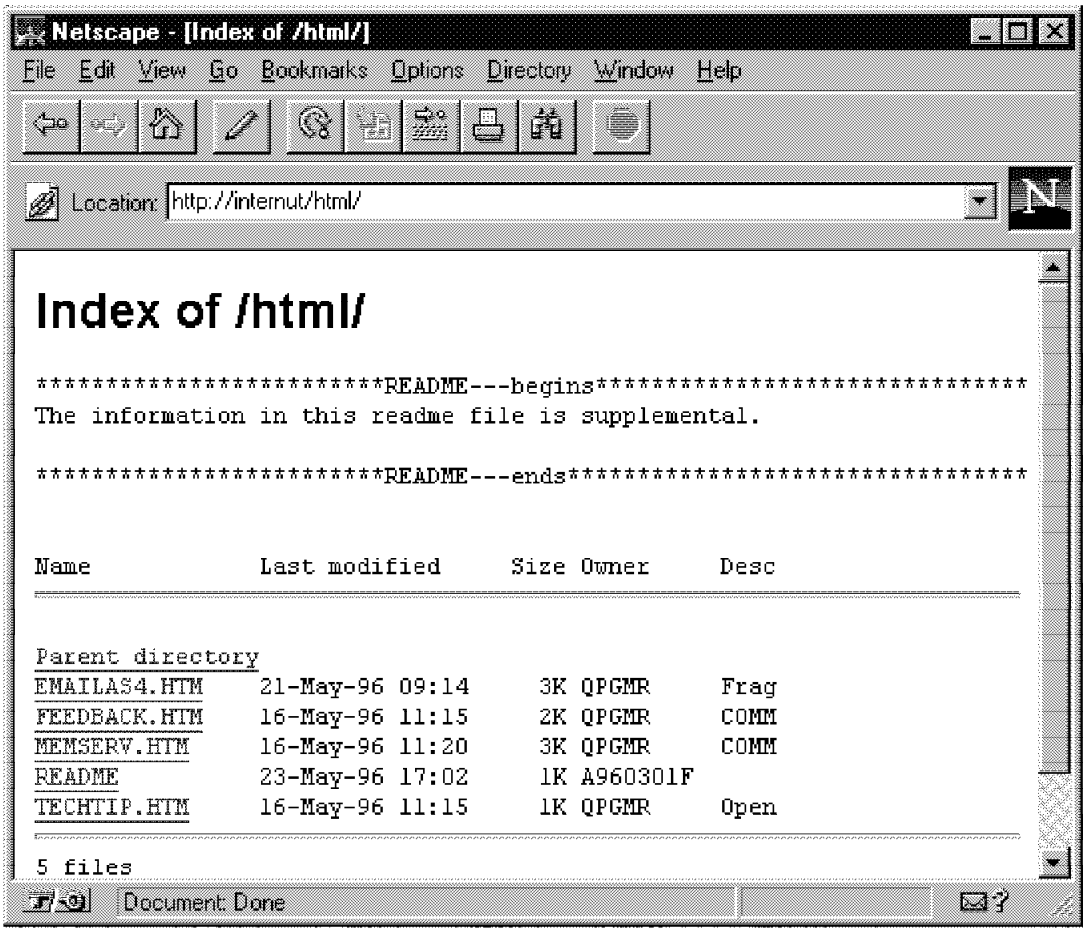


Figure 108. Using AS/400 HTTP Server's Directory Access

When an index of the directory is shown and the directory contains a file called *README*, the file may be imbedded to the index. The position of the readme file in the index is mandated by the HTTP configuration. A selected file either is shown to you by the browser or handled in some other way depending on your browser's settings. All in all, the file is handled in the same way as if it had been retrieved using a normal web page link.

6.6 Practical FTP Scenarios

The following sections contain programming examples. The first example is an automated FTP batch application. It uses quite a different approach than the examples in previous red books. The second example contains FTP exit programs.

All *programs and source files* are located in the library *ITSOIC400*.

6.6.1 Automated Batch

FTPBATCH is a small application for transferring files in an automated manner using FTP. It consists of one command, one CL program, and two RPG programs. The application can be easily integrated to existing applications.

Command FTPBATCH is the interface to other applications. Because it returns a parameter value, it must be used from a CL program or REXX procedure. The command processing program for FTPBATCH is STRFTP1C.

```
/*****************************************************************/
/*                                                               */
/*                        ** NOTE **                             */
/*                                                               */
/* This material contains programming source code for your       */
/* consideration. These examples have not been thoroughly tested */
/* under all conditions. IBM, therefore, cannot guarantee or imply */
/* reliability, serviceability, performance or function of these */
/* programs.  All programs contained herein are provided "AS IS". */
/* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A   */
/* PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                  */
/*                                                               */
/*****************************************************************/
 FTPBATCH:   CMD        PROMPT('START A FTP BATCH TRANSFER')
             PARM       KWD(RMTSYS) TYPE(*CHAR) LEN(50) MIN(1) +
                          PROMPT('Remote system for FTP transfer')
             PARM       KWD(LCLFIL) TYPE(*CHAR) LEN(50) MIN(1) +
                          PROMPT('Local file''s name')
             PARM       KWD(RMTFIL) TYPE(*CHAR) LEN(50) MIN(1) +
                          PROMPT('Remote file''s name')
             PARM       KWD(FUNCTION) TYPE(*CHAR) LEN(1) RSTD(*YES) +
                          VALUES(P G) SPCVAL((p P) (g G)) MIN(1) +
                          PROMPT('Function (Put/Get)')
             PARM       KWD(USR) TYPE(*CHAR) LEN(10) MIN(1) +
                          PROMPT('User profile in remote system')
             PARM       KWD(PWD) TYPE(*CHAR) LEN(10) MIN(1) +
                          PROMPT('Password to remote system')
             PARM       KWD(CMD1) TYPE(*CHAR) LEN(20) DFT( ) +
                          PROMPT('User defined FTP command 1')
             PARM       KWD(CMD2) TYPE(*CHAR) LEN(20) DFT( ) +
                          PROMPT('User defined FTP command 2')
             PARM       KWD(CMD3) TYPE(*CHAR) LEN(20) DFT( ) +
                          PROMPT('User defined FTP command 3')
             PARM       KWD(SUCCES) TYPE(*CHAR) LEN(1) RTNVAL(*YES) +
                          PROMPT('1=Transfer successful / 0=Not')
```

Figure 109. Command FTPBATCH

The main program for the application is STRFTP1C. It receives as parameters:

- IP address or host name of remote system.
- Local name for the file to be transferred.
- Remote name for the file to be transferred.
- Transfer function to be executed (Put or Get).
- User ID for the remote system.
- Password for the remote system.
- Three FTP commands to be executed before actual transfer.

STRFTP1C returns as a parameter:

- An indicator of the success of transfer

Notes (See Figure 110 on page 127.)

1 STRFTP1C calls program BLDFTP1R to create a file containing all of the needed FTP commands.

2 STRFTP1C calls program CHKFTP1R to check the result of the transfer.

3 All work files are created in the library QTEMP.

4 FTP is run using system defaults. If you need to use special conversion tables for the transfer, modify the parameters of STRTCPFTP command in the program.

```
/*********************************************************************/
/*                                                                   */
/*                     PROGRAM FUNCTION                              */
/*                                                                   */
/* This is the CPP for command FTPBATCH.                             */
/*                                                                   */
/* It performs a batch FTP transfer of one file with a TCP/IP        */
/* network connected FTP server.                                     */
/*                                                                   */
/* Program parameters:                                               */
/*                                                                   */
/*        &TGTSYS    (in )   -   Target system                       */
/*        &LCLFIL    (in )   -   Local file name                     */
/*        &RMTFIL    (in )   -   Remote file name                    */
/*        &FUNCTION  (in )   -   Function to be executed             */
/*                           values: 'P' = Put                       */
/*                                   'G' = Get                       */
/*        &USR       (in )   -   User ID in remote system            */
/*        &PWD       (in )   -   Password in remote system           */
/*        &CMD1      (in )   -   Optional FTP subcommand             */
/*                               (Executed before transfer)          */
/*        &CMD2      (in )   -   Optional FTP subcommand             */
/*                               (Executed before transfer)          */
/*        &CMD3      (in )   -   Optional FTP subcommand             */
/*                               (Executed before transfer)          */
/*        &SUCCES    (out)   -   Success indicator                   */
/*                           values: '0' = Transfer failed           */
/*                                   '1' = Transfer successful       */
/*                                                                   */
/* All work files are created into library QTEMP.                    */
/*                                                                   */
/* Programs called:                                                  */
/*                                                                   */
/*    1   BLDFTP1R (ILE RPG) -   Creates the FTP script              */
/*                               for transfer                        */
/*    2   CHKFTP1R (ILE RPG) -   Checks the success of transfer      */
/*                               by reading the FTP transfer log     */
/*                                                                   */
/*********************************************************************/
/*                                                                   */
/*                        ** NOTE **                                 */
/*                                                                   */
/* This material contains programming source code for your           */
/* consideration. These examples have not been thoroughly tested     */
/* under all conditions. IBM, therefore, cannot guarantee or imply   */
/* reliability, serviceability, performance or function of these     */
/* programs.  All programs contained herein are provided "AS IS".    */
/* THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A       */
/* PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                      */
/*                                                                   */
/*********************************************************************/
```

Figure 110 (Part 1 of 3). CL Program STRFTP1C

```
            PGM        PARM(&TGTSYS   +
                            &LCLFIL   +
                            &RMTFIL   +
                            &FUNCTION +
                            &USR      +
                            &PWD      +
                            &CMD1     +
                            &CMD2     +
                            &CMD3     +
                            &SUCCES   )

/*****************************************************************/
/* Parameters                                                    */
/*****************************************************************/
            DCL        VAR(&TGTSYS  ) TYPE(*CHAR) LEN(50)
            DCL        VAR(&LCLFIL  ) TYPE(*CHAR) LEN(50)
            DCL        VAR(&RMTFIL  ) TYPE(*CHAR) LEN(50)
            DCL        VAR(&FUNCTION) TYPE(*CHAR) LEN(1 )
            DCL        VAR(&USR     ) TYPE(*CHAR) LEN(10)
            DCL        VAR(&PWD     ) TYPE(*CHAR) LEN(10)
            DCL        VAR(&CMD1    ) TYPE(*CHAR) LEN(20)
            DCL        VAR(&CMD2    ) TYPE(*CHAR) LEN(20)
            DCL        VAR(&CMD3    ) TYPE(*CHAR) LEN(20)
            DCL        VAR(&SUCCES  ) TYPE(*CHAR) LEN(1)

/*****************************************************************/
/* Local variables                                               */
/*****************************************************************/
            DCL        VAR(&OKCODE  ) TYPE(*CHAR) LEN(3)
            DCL        VAR(&ON      ) TYPE(*CHAR) LEN(1) VALUE('1')
            DCL        VAR(&OFF     ) TYPE(*CHAR) LEN(1) VALUE('0')

/*****************************************************************/
/* Delete/Create the FTP script file                             */
/*****************************************************************/
       3    DLTF       FILE(QTEMP/FTPCMD)
            MONMSG     MSGID(CPF0000)
            CRTSRCPF   FILE(QTEMP/FTPCMD) MBR(FTP)

/*****************************************************************/
/* Build the FTP script                                          */
/*****************************************************************/
       1    OVRDBF     FILE(FTPCMD) TOFILE(QTEMP/FTPCMD) MBR(FTP)
            CALL       PGM(BLDFTP1R) PARM(&USR &PWD &FUNCTION +
                                          &LCLFIL &RMTFIL &CMD1 &CMD2 &CMD3)

/*****************************************************************/
/* Delete/Create the FTP transfer log file                       */
/*****************************************************************/
       3    DLTF       FILE(QTEMP/FTPLOG)
            MONMSG     MSGID(CPF0000)
            CRTPF      FILE(QTEMP/FTPLOG) RCDLEN(132) MBR(FTPLOG)

/*****************************************************************/
/* Execute FTP transfer                                          */
/*****************************************************************/
            OVRDBF     FILE(INPUT ) TOFILE(QTEMP/FTPCMD) MBR(FTP)
            OVRDBF     FILE(OUTPUT) TOFILE(QTEMP/FTPLOG) MBR(FTPLOG)
       4    STRTCPFTP  RMTSYS(&TGTSYS)
            DLTOVR     FILE(*ALL)
```

Figure 110 (Part 2 of 3). CL Program STRFTP1C

```
/*********************************************************************/
/* Check FTP transfer log file to find out if the transfer was      */
/* successful                                                        */
/*********************************************************************/
       2  CALL       PGM(CHKFTP1R) PARM(&OKCODE)
          IF         COND(&OKCODE *EQ 'YES') THEN(CHGVAR +
                       VAR(&SUCCES) VALUE(&ON))
          ELSE       CMD(CHGVAR VAR(&SUCCES) VALUE(&OFF))

          ENDPGM
```

Figure 110 (Part 3 of 3). CL Program STRFTP1C

Program BLDFTP1R is called by STRFTP1C. It writes into a source physical file the FTP command needed to execute the file transfer. It receives as parameters:

- User ID for the remote system.
- Password for the remote system.
- Transfer function to be executed (Put or Get).
- Local name for the file to be transferred.
- Remote name for the file to be transferred.
- Three FTP commands to be executed before actual transfer.

Notes (See Figure 111 on page 130.)

1 File names need to contain a full path unless you use optional commands to change current directories to point into correct directories/libraries.

2 Optional commands are not checked to contain valid FTP commands and are added to the file if they are non-blank.

3 The replace option is always used with the GET command. You can modify the application to make this optional

```
F/TITLE CREATE FTP SCRIPT
 ********************************************************************
 *                                                                  *
 *                      PROGRAM FUNCTION                            *
 *                                                                  *
 * This program creates the FTP script for FTPBATCH command.        *
 *                                                                  *
 ********************************************************************
 *                                                                  *
 *                         ** NOTE **                               *
 *                                                                  *
 * This material contains programming source code for your          *
 * consideration. These examples have not been thoroughly tested    *
 * under all conditions. IBM, therefore, cannot guarantee or imply  *
 * reliability, serviceability, performance or function of these    *
 * programs.  All programs contained herein are provided "AS IS".   *
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A      *
 * PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                     *
 *                                                                  *
 ********************************************************************
F/SPACE 3
 ********************************************************************
 *                                                                  *
 *                       INDICATOR USAGE                            *
 *                                                                  *
 *   IND.  DESCRIPTION                                             *
 *                                                                  *
 *    LR - CLOSE FILES ON EXIT                                      *
 *                                                                  *
 ********************************************************************
F/EJECT
 ********************************************************************
 *FILES USED BY THIS PROGRAM
 ********************************************************************
FFTPCMD    O    F   92        DISK                                    FTP script file
D/EJECT
 ********************************************************************
 * DATA STRUCTURES USED BY THIS PROGRAM
 ********************************************************************
D/SPACE 2
 *
 * Source file record format
 *
D CmdRec          DS
D  SRCSEQ                 1      6  2
D  SRCDAT                 7     12  0
D  SRCDTA                13     92
```

Figure 111 (Part 1 of 4). ILE RPG Program BLDFTP1R

```
C/EJECT
 ****************************************************************
 * VARIABLE DEFINITIONS AND LISTS USED BY THIS PROGRAM          *
 ****************************************************************
C/SPACE 2
 *
 * Define parameter list
 *
C     *ENTRY        PLIST
C                   PARM                    RMTUSR          10          Remote user ID
C                   PARM                    RMTPWD          10          Remote password
C                   PARM                    FUNCTION         1          Function to execute:
 *                                                    possible values:  P = PUT
 *                                                                      G = GET
C           1       PARM                    LCLFIL          50          Local file name
 *                                                                      (fully qualified)
C           1       PARM                    RMTFIL          50          Remote file name
 *                                                                      (fully qualified)
C           2       PARM                    CMD1            20          1st optional FTP cmd
C           2       PARM                    CMD2            20          2nd optional FTP cmd
C           2       PARM                    CMD3            20          3rd optional FTP cmd
C/EJECT
 ****************************************************************
 * The Main Program                                             *
 ****************************************************************
 *
C                   EXSR      CFOO1
 *
C                   EVAL      *INLR = *ON
C                   RETURN
C*
C/EJECT
 ****************************************************************
 * S U B R O U T I N E S                                        *
 ****************************************************************
 ****************************************************************
 * Main Processing                                              *
 ****************************************************************
C     CFOO1         BEGSR
 *
C                   Z-ADD     *ZERO         SEQ              6 2
 *
C                   EXSR      CFOO2
C                   EXSR      CFOO3
C                   EXSR      CFOO4
C                   EXSR      CFOO5
 *
C                   ENDSR
C/EJECT
 ****************************************************************
 * Build and Write User ID and Password record                  *
 ****************************************************************
C     CFOO2         BEGSR
 *
C                   Z-ADD     *ZERO         SRCDAT
C                   ADD       1             SEQ
C                   Z-ADD     SEQ           SRCSEQ
C     RMTUSR        CAT(P)    RMTPWD:1      SRCDTA
C                   WRITE     FTPCMD        CmdRec
 *
C                   ENDSR
```

Figure 111 (Part 2 of 4). ILE RPG Program BLDFTP1R

```
C/EJECT
 ********************************************************************
 * Write Optional FTP command records                               *
 ********************************************************************
C     CFOO3         BEGSR
 *
C  2  CMD1          IFNE      *BLANK
C                   Z-ADD     *ZERO         SRCDAT
C                   ADD       1             SEQ
C                   Z-ADD     SEQ           SRCSEQ
C                   MOVEL(P)  CMD1          SRCDTA
C                   WRITE     FTPCMD        CmdRec
C                   ENDIF
 *
C  2  CMD2          IFNE      *BLANK
C                   Z-ADD     *ZERO         SRCDAT
C                   ADD       1             SEQ
C                   Z-ADD     SEQ           SRCSEQ
C                   MOVEL(P)  CMD2          SRCDTA
C                   WRITE     FTPCMD        CmdRec
C                   ENDIF
 *
C  2  CMD3          IFNE      *BLANK
C                   Z-ADD     *ZERO         SRCDAT
C                   ADD       1             SEQ
C                   Z-ADD     SEQ           SRCSEQ
C                   MOVEL(P)  CMD3          SRCDTA
C                   WRITE     FTPCMD        CmdRec
C                   ENDIF
 *
C                   ENDSR
C/EJECT
 ********************************************************************
 * Build and Write PUT/GET Command Record                           *
 ********************************************************************
C     CFOO4         BEGSR
 *
C                   Z-ADD     *ZERO         SRCDAT
C                   ADD       1             SEQ
C                   Z-ADD     SEQ           SRCSEQ
 *
C                   SELECT
C     FUNCTION      WHENEQ    'P'                                     PUT function
C     'PUT'         CAT(P)    LCLFIL:1      SRCDTA
C     SRCDTA        CAT(P)    RMTFIL:1      SRCDTA
C                   WRITE     FTPCMD        CmdRec
 *
C     FUNCTION      WHENEQ    'G'                                     GET function
C     'GET'         CAT(P)    RMTFIL:1      SRCDTA
C     SRCDTA        CAT(P)    LCLFIL:1      SRCDTA
C  3  SRCDTA        CAT(P)    '(Replace':1  SRCDTA                  We use in this
 *                                                                   example always
 *                                                                   REPLACE option in
 *                                                                   GET. Can be made
 *                                                                   optional.
C                   WRITE     FTPCMD        CmdRec
C                   ENDSL
 *
C                   ENDSR
```

Figure 111 (Part 3 of 4). ILE RPG Program BLDFTP1R

```
C/EJECT
 ********************************************************************
 * Build and Write QUIT Record                                      *
 ********************************************************************
C     CFOO5         BEGSR
 *
C                   Z-ADD     *ZERO         SRCDAT
C                   ADD       1             SEQ
C                   Z-ADD     SEQ           SRCSEQ
C                   MOVEL(P)  'QUIT'        SRCDTA
C                   WRITE     FTPCMD        CmdRec
 *
C                   ENDSR
```

Figure 111 (Part 4 of 4). ILE RPG Program BLDFTP1R

Program CHKFTP1R is called by STRFTP1C. It reads through the physical file used as an FTP log to find out the result of the file transfer. It returns as a parameter:

- An indicator of the success of transfer.

Notes (See Figure 112.)

1 FTP reply codes 226 and 250 are used by FTP servers to indicate the end of transfer. FTP reply code 250 is used, for example, in transfers between two AS/400 systems.

2 Because FTP reply code 250 is used also for other functions, some of the explanatory part of the message has to be checked.

```
F/TITLE CHECK FTP LOG
 ********************************************************************
 *                                                                  *
 *                      PROGRAM FUNCTION                            *
 *                                                                  *
 * This program reads through the FTP transfer log file and         *
 * tries to find out if the transfer was successful.                *
 *                                                                  *
 ********************************************************************
 *                                                                  *
 *                          ** NOTE **                              *
 *                                                                  *
 * This material contains programming source code for your          *
 * consideration. These examples have not been thoroughly tested    *
 * under all conditions. IBM, therefore, cannot guarantee or imply  *
 * reliability, serviceability, performance or function of these    *
 * programs.  All programs contained herein are provided "AS IS".   *
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A      *
 * PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                     *
 *                                                                  *
 ********************************************************************
F/SPACE 3
 ********************************************************************
 *                                                                  *
 *                       INDICATOR USAGE                            *
 *                                                                  *
 *   IND.  DESCRIPTION                                             *
 *                                                                  *
 *    99 - FTPLOG EOF                                               *
 *    LR - CLOSE FILES ON EXIT                                      *
 *                                                                  *
 ********************************************************************
```

Figure 112 (Part 1 of 2). ILE RPG Program CHKFTP1R

```
F/EJECT
 ****************************************************************
 *FILES USED BY THIS PROGRAM
 ****************************************************************
FFTPLOG    IF   F  132        DISK                                    Transfer log fil
F/EJECT
 ****************************************************************
 * DATA STRUCTURES USED BY THIS PROGRAM                         *
 ****************************************************************
 *
 * Define constants
 *
D FileXfer        C                   CONST('File transfer')
 *
 * Define input record format
 *
D LogRec          DS            132
D  ReplyCode                1      3
D  ExtraInfo                5     17
C/EJECT
 ****************************************************************
 * VARIABLE DEFINITIONS AND LISTS USED BY THIS PROGRAM          *
 ****************************************************************
C/SPACE 2
C     *ENTRY        PLIST
 * Return parameters:
C                   PARM                    OK                3
 *                                                    possible values:   NOT = Not successfull
 *                                                                       YES = Successfull xfer
C/EJECT
 ****************************************************************
 * The Main Program                                             *
 ****************************************************************
 *
C                   MOVE      'NOT'         OK
 *
C     *IN99         DOWEQ     *OFF
C                   READ      FTPLOG        LogRec                  99
C     *IN99         IFEQ      *OFF
 *
C                   SELECT
 * Remote system was something else than an AS/400 or any other that uses 250 as a response
C  1  ReplyCode     WHENEQ    '226'
C                   MOVE      'YES'         OK
C                   LEAVE
 *
 * Remote system was an AS/400 or any other that uses 250 as a response
C  1  ReplyCode     WHENEQ    '250'
C  2  ExtraInfo     ANDEQ     FileXfer
C                   MOVE      'YES'         OK
C                   LEAVE
 *
C                   ENDSL
C                   ENDIF
C                   ENDDO
 *
C                   EVAL      *INLR = *ON
C                   RETURN
C/EJECT
 ****************************************************************
 *                                                              *
 * ...and here be the dragons.                                  *
 *                                                              *
 ****************************************************************
```

Figure 112 (Part 2 of 2). ILE RPG Program CHKFTP1R

The application was tested using the following CL program.

```
PGM
DCL        VAR(&RESULT) TYPE(*CHAR) LEN(1) VALUE('0')
ITSOIC400/FTPBATCH   RMTSYS(SYSTEM01) +
             LCLFIL('/QSYS.LIB/hessu.lib/qclsrc.file/te.+
             mbr') +
             RMTFIL('/QSYS.LIB/hessu.lib/qcl.file/te.mbr+
             ') FUNCTION(P) USR(HPMA) PWD(SUNDANCE) +
             CMD1('Namefmt 1') SUCCES(&RESULT)
IF         COND(&RESULT = '1') THEN(SNDMSG MSG('OK') +
             TOUSR(*REQUESTER))
ELSE       CMD(SNDMSG MSG('NOT OK') TOUSR(*REQUESTER))
ENDPGM
```

Figure 113. Test Program for FTPBATCH

And after running successfully and transferring one source file member from one AS/400 system to another, the contents of the work files are in the following example.

```
 ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8
HPMA SUNDANCE
Namefmt 1
PUT /QSYS.LIB/hessu.lib/qclsrc.file/te.mbr /QSYS.LIB/hessu.lib/qcl.file/te.mbr 1
QUIT
```

Figure 114. FTP Commands from FTPBATCH

Note

1 When you are using NAMEFMT 1 file names, notice that the maximum length for one FTP command, in this case, is 80 characters. You might want to use one or two of the optional FTP commands to change the current directories to proper ones.

```
Output redirected to a file.
Input read from specified override file.
Connecting to host SYSTEM01 at address 10.180.128.3 using port 21.
220-QTCP at SYSTEM01.
220 Connection will close if idle more than 60 minutes.
 OS/400 is the remote operating system. The TCP/IP version is "V3R1M0".
Enter login ID (a960101f):
331 Enter password.
230 HPMA logged on.
250  Now using naming format "0".
257 "QGPL" is current library.
Enter an FTP subcommand.
> Namefmt 1
250  Now using naming format "1".
Server NAMEFMT is 1.
Client NAMEFMT is 1.
Enter an FTP subcommand.
> PUT /QSYS.LIB/hessu.lib/qclsrc.file/te.mbr /QSYS.LIB/hessu.lib/qcl.file/te.mbr
200 PORT subcommand request successful.
150 Sending file to member TE in file QCL in library HESSU.
250 File transfer completed successfully.
643 bytes transferred in 2.522 seconds. Transfer rate 0.255 KB/sec.
Enter an FTP subcommand.
> QUIT
221 QUIT subcommand received.
```

Figure 115. FTP Log from FTPBATCH

6.6.2 FTP Exit Programs (ILE RPG)

This section shows us some example FTP exit programs that are a good start for your study and understanding of the FTP exit points. The first section shows us a sample FTP logon exit program and the second show us a request validation program.

6.6.2.1 FTP Logon Exit Program (ILE RPG)

This FTP logon exit program enforces the following security scheme.

- Anonymous FTP logon is accepted and user profile ANONYMOUS is used for this.
- Only anonymous FTP logons are logged.
- Users whose IP addresses belong to network 10.xxx.xxx.xxx are not allowed to do an FTP logon with their own user profile, but they can do anonymous logons.

Note (See Figure 116 on page 137.)

1 User ID "ANONYMOUS" is always supplied to the exit program in upper case.

```
 ********************************************************************
 *                                                                  *
 *                       PROGRAM FUNCTION                           *
 *                                                                  *
 * This program demonstrates some of the abilities an FTP Server    *
 * Logon Exit Program can have.                                     *
 *                                                                  *
 ********************************************************************
 *                                                                  *
 *                          ** NOTE **                              *
 *                                                                  *
 * This material contains programming source code for your          *
 * consideration. These examples have not been thoroughly tested    *
 * under all conditions. IBM, therefore, cannot guarantee or imply  *
 * reliability, serviceability, performance or function of these    *
 * programs.  All programs contained herein are provided "AS IS".   *
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A      *
 * PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                     *
 *                                                                  *
 ********************************************************************
F/SPACE 3
 ********************************************************************
 *                                                                  *
 *                       INDICATOR USAGE                            *
 *                                                                  *
 *   IND.  DESCRIPTION                                             *
 *                                                                  *
 *    LR - CLOSE FILES ON EXIT                                      *
 *                                                                  *
 ********************************************************************
F/EJECT
 ********************************************************************
 * DATA STRUCTURES USED BY THIS PROGRAM                             *
 ********************************************************************
 *
 * Define constants
 *
1 D Anonym          C                   CONST('ANONYMOUS ')
  D Text1           C                   CONST('Anonymous (')
  D Text2           C                   CONST(') FTP logon')
  D InvalidNet      C                   CONST('10.')
C/EJECT
 ********************************************************************
 * VARIABLE DEFINITIONS AND LISTS USED BY THIS PROGRAM              *
 ********************************************************************
C/SPACE 2
 *
 * Define binary parameters
 *
D                 DS
D  APPIDds                1      4B 0
D  USRLENds               5      8B 0
D  AUTLENds               9     12B 0
D  IPLENds               13     16B 0
D  RETCDds               17     20B 0
 *
C     *LIKE         DEFINE    APPIDds       APPIDIN
C     *LIKE         DEFINE    USRLENds      USRLENIN
C     *LIKE         DEFINE    AUTLENds      AUTLENIN
C     *LIKE         DEFINE    IPLENds       IPLENIN
C     *LIKE         DEFINE    RETCDds       RETCDOUT
```

Figure 116 (Part 1 of 3). RPG Logon Exit Program (FTPLOGON)

```
 *
 * Define parameter list
 *
C     *Entry        PLIST
 * Input parameters:
C                   PARM                    APPIDIN                        Application ID
 *                                                     possible values:   1 = FTP Server Program
C                   PARM                    USRIN           999            User ID
C                   PARM                    USRLENIN                       Length of User ID
C                   PARM                    AUTIN           999            Authentication Strg
C                   PARM                    AUTLENIN                       Length of Auth. Strg
C                   PARM                    IPADDRIN         15            Client IP Address
C                   PARM                    IPLENIN                        Length of IP Address
 * Return parameters:
C                   PARM                    RETCDOUT                       Return Code (Out)
 *                                                     possible values:   0 = Reject Logon
 *                                                                        1 = Continue Logon
 *                                                                        2 = Continue Logon,
 *                                                                            override current
 *                                                                            library
 *                                                                        3 = Continue Logon,
 *                                                                            override user prf,
 *                                                                            password
 *                                                                        4 = Continue Logon,
 *                                                                            override user prf,
 *                                                                            password, current
 *                                                                            library
 *                                                                        5 = Accept logon with
 *                                                                            user prf returned
 *                                                                        6 = Accept logon with
 *                                                                            user prf returned,
 *                                                                            override current
 *                                                                            library
C                   PARM                    USRPRFOUT       10             User Profile (Out)
C                   PARM                    PASSWDOUT       10             Password (Out)
C                   PARM                    CURLIBOUT       10             Current Lib. (Out)
C/EJECT
 ********************************************************************
 * THE MAIN PROGRAM                                                 *
 ********************************************************************
 *
 * Check for ANONYMOUS user
 *    1
C     USRLENIN      SUBST(P)  USRIN:1       User            10
C     User          IFEQ      Anonym
C                   MOVEL     Anonym        USRPRFOUT
 *
 * Check if the user entered something as a e-mail address
 *
C     AUTLENIN      IFGT      *ZERO                                        E-mail addr. entered
 *
 * Check if the E-mail address is a valid one
 *
C                   Z-ADD     0             i                3 0
C     '@'           SCAN      AUTIN:1       i                             Valid E-mail address
 *                                                                         contains @ character
 *
C     i             IFGT      0                                           Found a '@'
C     AUTLENIN      SUBST(P)  AUTIN:1       Email           30
C                   Z-ADD     5             RETCDOUT                      Accept Logon
```

Figure 116 (Part 2 of 3). RPG Logon Exit Program (FTPLOGON)

```
 *
 * Log Anonymous FTP Logon to message queue QSYSOPR
 * (The logging should be done to a secure physical file!!!!!!!)
 *
C     Text1         CAT(p)    Email:0       Message          43
C     Message       CAT(p)    Text2:0       Message
C     Message       DSPLY     'QSYSOPR'
 *
C                   ELSE                                                 Invalid E-mail addr
C                   Z-ADD     0             RETCDOUT                     Reject Logon attempt
C                   ENDIF
 *
C                   ELSE                                                 No E-mail address
C                   Z-ADD     0             RETCDOUT                     Reject Logon attempt
C                   ENDIF
 *
C                   ELSE
 *
 * Any Other User: Proceed with Normal Logon Processing, but the Client address must not belong
 *                 to network 10.xxx.xxx.xxx
 *
C     3             SUBST     IPADDRIN:1    TheNet            3
C     TheNet        IFEQ      InvalidNet                                 Wrong Net
C                   Z-ADD     0             RETCDOUT                     Reject Logon attempt
C                   ELSE                                                 Right Net
C                   Z-ADD     1             RETCDOUT                     Continue with Logon
C                   ENDIF
 *
C                   ENDIF
 *
C                   EVAL      *INLR = *ON
C                   RETURN
```

Figure 116 (Part 3 of 3). RPG Logon Exit Program (FTPLOGON)

6.6.2.2 FTP Request Validation Program (ILE RPG)

The following program is a combined FTP client and server request validation program. It enforces the following security scheme for the FTP client.

- User "JOEBAD" is not allowed to do anything.
- User "JOENORMAL" is allowed to transfer files from directory /itsoic.400 and library ITSOIC400 to the server.
- User "JOEGOOD" with few others is allowed to everything.
- All the other users are not allowed to do anything.
- No FTP requests are logged.

It enforces the following security scheme for the FTP server.

- Anonymous FTP user (has been accepted with user profile ANONYMOUS) is allowed to transfer files from directory /itsoic.400 and library ITSOIC400 to the client.
- Other users are limited in no way.
- No FTP requests are logged.

A hidden function in the server part of the program causes one save file to be refreshed whenever it is requested by an FTP client.

Note (See Figure 117 on page 140.)

1 Path and file names, except QSYS.LIB names, are given to the program in the same format as the user wrote them. To ease comparison with allowed directory names, they should be converted to upper or lower

case. This must not be done to QOpenSys path names, which are case sensitive.

2 During the writing of this book and testing of the features of Internet Connection for AS/400, we noticed that this exit program did too good of a job securing our system! To get any real work done we had to leave a back door open in the security scheme.

Note: Please change this part of the program if you want to close the security exposure!

```
 ****************************************************************
 *                                                              *
 *                     PROGRAM FUNCTION                         *
 *                                                              *
 * This program demonstrates some of the abilities an FTP Server*
 * Client and Server Request Validation Exit Program can have.  *
 *                                                              *
 ****************************************************************
 *                                                              *
 *                        ** NOTE **                            *
 *                                                              *
 * This material contains programming source code for your      *
 * consideration. These examples have not been thoroughly tested*
 * under all conditions. IBM, therefore, cannot guarantee or imply*
 * reliability, serviceability, performance or function of these*
 * programs.  All programs contained herein are provided "AS IS".*
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A  *
 * PARTICULAR PURPOSE ARE EXPRESSLY DISCLAIMED.                 *
 *                                                              *
 ****************************************************************
F/SPACE 3
 ****************************************************************
 *                                                              *
 *                     INDICATOR USAGE                          *
 *                                                              *
 *   IND.  DESCRIPTION                                          *
 *                                                              *
 *    LR - CLOSE FILES ON EXIT                                  *
 *                                                              *
 ****************************************************************
F/EJECT
 ****************************************************************
 * DATA STRUCTURES USED BY THIS PROGRAM                         *
 ****************************************************************
 *
 * Define constants
 *
D Anonym          C                   CONST('ANONYMOUS ')
D PublicLib       C                   CONST('/QSYS.LIB/ITSOIC400.LIB')
D PublicDir       C                   CONST('//ITSOIC.400')
 *
 * Some CL commands to used later on in the program
 *
D ClearSavf       C                   CONST('CLRSAVF ITSOIC400/TURVIS')
D SaveLib         C                   CONST('SAVLIB LIB(ITSOIC400) -
D                                     DEV(*SAVF) -
D                                     SAVF(ITSOIC400/TURVIS)')
```

Figure 117 (Part 1 of 7). RPG Request Validation Program (FTPRQSVLD)

```
 *
 * A value to be used to trigger a benevolent 'Trojan Horse'
 *
D Savetti         C                   CONST('ITSOIC400.LIB/TURVIS.FILE')   Extension is FILE
 *                                                                        although it is a
 *                                                                        SAVF (and entered as
 *                                                                        SAVF by the user)
 *
 * Some nice fields to help us through from lower to upper case character conversion
 *    1
D LW              C                   CONST('abcdefghijklmnopqrstuvwxyz')
D UP              C                   CONST('ABCDEFGHIJKLMNOPQRSTUVWXYZ')
 *
D NeverAllow      C                   CONST(-1)
D DontAllow       C                   CONST(0)
D Allow           C                   CONST(1)
D AlwaysAllw      C                   CONST(2)
C/EJECT
 ********************************************************************
 * VARIABLE DEFINITIONS AND LISTS USED BY THIS PROGRAM              *
 ********************************************************************
C/SPACE 2
 *
 * Define binary parameters
 *
D                 DS
D  APPIDds                1      4B 0
D  OPIDds                 5      8B 0
D  IPLENds                9     12B 0
D  OPLENds               13     16B 0
D  ALLOWOPds             17     20B 0
 *
C     *LIKE         DEFINE    APPIDds       APPIDIN
C     *LIKE         DEFINE    OPIDds        OPIDIN
C     *LIKE         DEFINE    IPLENds       IPLENIN
C     *LIKE         DEFINE    OPLENds       OPLENIN
C     *LIKE         DEFINE    ALLOWOPds     ALLOWOP
 *
C     *LIKE         DEFINE    OPINFOIN      OPINFO
```

Figure 117 (Part 2 of 7). RPG Request Validation Program (FTPRQSVLD)

```
 *
 * Define parameter list
 *
C     *Entry        PLIST
 * Input parameters:
C                   PARM                    APPIDIN                        Application ID
 *                                                     possible values:   0 = FTP Client Program
 *                                                                        1 = FTP Server Program
C                   PARM                    OPIDIN                         Operation ID
 *                                                     possible values:   0 = Initialize Session
 *                                                                        1 = Create Dir/Lib
 *                                                                        2 = Delete Dir/Lib
 *                                                                        3 = Set Current Dir
 *                                                                        4 = List Dir/Lib
 *                                                                        5 = Delete Files
 *                                                                        6 = Send Files
 *                                                                        7 = Receive Files
 *                                                                        8 = Rename Files
 *                                                                        9 = Execute CL cmd
C                   PARM                    USRPRF          10             User Profile
C                   PARM                    IPADDRIN        15             Remote IP Address
C                   PARM                    IPLENIN                        Length of IP Address
C                   PARM                    OPINFOIN        999            Operation-spec. Info
C                   PARM                    OPLENIN                        Length of Oper. Spec
 * Return parameter:
C                   PARM                    ALLOWOP                        Allow Operation (Out
 *                                                     possible values:  -1 = Never Allow
 *                                                                           (And don't bother
 *                                                                           me with this ops
 *                                                                           in this session)
 *                                                                        0 = Reject Operation
 *                                                                        1 = Allow Operation
 *                                                                        2 = Always Allow Oper.
 *                                                                           (And don't bother
 *                                                                           me with this ops
 *                                                                           in this session)
C/EJECT
 ********************************************************************
 * The Main Program                                                 *
 ********************************************************************
 *
C                   SELECT
C     APPIDIN       WHENEQ    0
C                   EXSR      ClientRqs
C     APPIDIN       WHENEQ    1
C                   EXSR      ServerRqs
C                   ENDSL
 *
C                   EVAL      *INLR = *ON
C                   RETURN
```

Figure 117 (Part 3 of 7). RPG Request Validation Program (FTPRQSVLD)

```
C/EJECT
 ********************************************************************
 * S U B R O U T I N E S                                            *
 ********************************************************************
 ********************************************************************
 * Here we handle all the FTP Client request validation             *
 ********************************************************************
C     ClientRqs     BEGSR
 *
 * Check user profile
 *
C                   SELECT
 *
 * Check for 'bad' users who are not allowed to do anything ever
 *
C     USRPRF        WHENEQ    'JOEBAD    '
 *
C                   Z-ADD     NeverAllow    ALLOWOP                        Ops not allowed
 *
 * Check for 'normal' users who are not allowed to do some things
 *
C     USRPRF        WHENEQ    'JOENORMAL '
 *
C                   SELECT
 *
C     OPIDIN        WHENEQ    0                                            New Connection
C                   Z-ADD     Allow         ALLOWOP
 *
C     OPIDIN        WHENEQ    1                                            Create Directory/Lib
C     OPIDIN        OREQ      2                                            Delete Directory/Lib
C     OPIDIN        OREQ      5                                            Delete Files
C     OPIDIN        OREQ      7                                            Receive Files from S
C     OPIDIN        OREQ      8                                            Rename files
C     OPIDIN        OREQ      9                                            Execute CL Commands
 *
C                   Z-ADD     NeverAllow    ALLOWOP                        Ops never allowed
 *
C     OPIDIN        WHENEQ    3                                            Set Current Dir
C     OPIDIN        OREQ      4                                            List Directory/Lib
C     OPIDIN        OREQ      6                                            Send Files to Server
 *
 * Extract library and directory names for comparison with allowed areas
 *
C     OPLENIN       IFGE      11
C     11            SUBST     OPINFOIN:1    Directory        11
C                   ELSE
C     OPLENIN       SUBST(P)  OPINFOIN:1    Directory
C                   ENDIF
C  1  LW:UP         XLATE     Directory     Directory
 *
C     OPLENIN       IFGE      23
C     23            SUBST     OPINFOIN:1    Library          23
C                   ELSE
C     OPLENIN       SUBST(P)  OPINFOIN:1    Library
C                   ENDIF
 *
C     Directory     IFEQ      PublicDir                                    Allowed Directory
C     Library       OREQ      PublicLib                                    or Library
C                   Z-ADD     Allow         ALLOWOP
C                   ELSE
C                   Z-ADD     DontAllow     ALLOWOP
C                   ENDIF
 *
C                   OTHER
C                   Z-ADD     DontAllow     ALLOWOP
C                   ENDSL
```

Figure 117 (Part 4 of 7). RPG Request Validation Program (FTPRQSVLD)

```
     *
     * Check for 'cool' users who are allowed to do everything
     *
    C     USRPRF        WHENEQ    'JOEGOOD   '
    C     USRPRF        OREQ      'A960101B  '
    C     USRPRF        OREQ      'A960101C  '
    C     USRPRF        OREQ      'A960101D  '
    C     USRPRF        OREQ      'A960101E  '
    C     USRPRF        OREQ      'A960101F  '
    C     USRPRF        OREQ      'A960101Z  '
     * Allow All FTP Operations
    C                   Z-ADD     AlwaysAllw    ALLOWOP
     *
2    * Any Other User: We leave the back door open and allow
     * all operations. If you want to use this program for securing
     * your system, then close this door!
     *
    C                   OTHER
    C                   Z-ADD     AlwaysAllw    ALLOWOP
    C***************    Z-ADD     NeverAllow    ALLOWOP
    C                   ENDSL
     *
    C                   ENDSR
    C/EJECT
     ********************************************************************
     * Here we handle all the FTP Server request validation             *
     ********************************************************************
    C     ServerRqs     BEGSR
     *
     * Check for ANONYMOUS user
     *
    C     USRPRF        IFEQ      Anonym
     *
    C                   SELECT
     *
    C     OPIDIN        WHENEQ    1                                      Create Directory/Lib
    C     OPIDIN        OREQ      2                                      Delete Directory/Lib
    C     OPIDIN        OREQ      5                                      Delete Files
    C     OPIDIN        OREQ      7                                      Receive Files from C
    C     OPIDIN        OREQ      8                                      Rename files
    C     OPIDIN        OREQ      9                                      Execute CL Commands
     *
    C                   Z-ADD     NeverAllow    ALLOWOP                  Ops never allowed
```

Figure 117 (Part 5 of 7). RPG Request Validation Program (FTPRQSVLD)

```
 *
C     OPIDIN        WHENEQ    3                                        Set Current Dir
C     OPIDIN        OREQ      4                                        List Directory/Lib
C     OPIDIN        OREQ      6                                        Send Files to Client
 *
 * Extract library and directory names for comparison with allowed areas
 *
C     OPLENIN       IFGE      11
C     11            SUBST     OPINFOIN:1    Directory        11
C                   ELSE
C     OPLENIN       SUBST(P)  OPINFOIN:1    Directory
C                   ENDIF
C   1 LW:UP         XLATE     Directory     Directory
 *
C     OPLENIN       IFGE      23
C     23            SUBST     OPINFOIN:1    Library          23
C                   ELSE
C     OPLENIN       SUBST(P)  OPINFOIN:1    Library
C                   ENDIF
 *
C     Directory     IFEQ      PublicDir                              Allowed Directory
C     Library       OREQ      PublicLib                              or Library
C                   Z-ADD     Allow         ALLOWOP
C                   ELSE
C                   Z-ADD     DontAllow     ALLOWOP
C                   ENDIF
 *
C                   OTHER
C                   Z-ADD     DontAllow     ALLOWOP
C                   ENDSL
 *
C                   ELSE
```

Figure 117 (Part 6 of 7). RPG Request Validation Program (FTPRQSVLD)

```
 *
 * Any Other User: Allow All FTP Operations
 *
C     OPIDIN        IFEQ      6                                          Send Files to Client
 *
 * If client issued GET for save file HESSU in library HESSU then we refresh the contents
 *
 *
C     LW:UP         XLATE     OPINFOIN      OPINFO
C                   Z-ADD     0             i                3 0
C     Savetti       SCAN      OPINFO:1      i
 *
C     i             IFGT      0
 *
 * We assume that the save file exits and here clear the save file
 *
C                   MOVEL(p)  ClearSavf     Cmd              80
C                   Z-ADD     19            Len              15 5
C                   CALL      'QCMDEXC'                               9999
C                   PARM                    Cmd
C                   PARM                    Len
 *
 * and here we save the library to the save file
 *
C                   MOVEL(p)  SaveLib       Cmd
C                   Z-ADD     46            Len
C                   CALL      'QCMDEXC'                               9999
C                   PARM                    Cmd
C                   PARM                    Len
C                   ENDIF
C                   ENDIF
 *
C                   Z-ADD     Allow         ALLOWOP
C                   ENDIF
 *
C                   ENDSR
```

Figure 117 (Part 7 of 7). RPG Request Validation Program (FTPRQSVLD)

6.6.2.3 Anonymous FTP Session Example

The following example is an anonymous FTP session to an AS/400 server. Programs that are introduced previously were used as FTP exit programs on the server.

```
ftp> open internut
Connected to internut
220-QTCP at INTERNUT.
220 Connection will close if idle more than 60 minutes.
User (internut:(none)): anonymous
331 Guest logon in process, send complete E-mail address as password.
Password:                 1
530 Log on attempt by user ANONYMOUS rejected.
Login failed.
ftp> user anonymous
331 Guest logon in process, send complete E-mail address as password.
Password:
230 Guest logon accepted, access restrictions apply.  2
ftp> pwd
257 "ITSOIC400" is current library.
ftp> quote site namefmt 1
250  Now using naming format "1".
ftp> pwd
257 "ITSOIC400.LIB" is current library.
ftp> cd /
550 Request rejected.   3
ftp> cd /public
250 Current directory changed to /public.
ftp> cd /qsys.lib/itsoic400.lib
250 Current library changed to ITSOIC400.
```

1 The password entered was not a valid E-mail address so the attempt was rejected.

2 The second password was accepted as an E-mail address, but the message mentions restrictions because of exit programs.

3 Only a public library and directory are allowed to an anonymous user. So the request to change the current directory to root failed due to the checking of the exit program.

Chapter 7. Gopher on the AS/400 System

The word Gopher is the name of the Internet protocol designed for distributed document search and retrieval. This protocol was originally developed at the University of Minnesota in 1991, and is a menu interface.

The Internet Gopher protocol is designed primarily to act as a distributed document delivery system. While documents (and services) reside on many servers, Gopher client software presents users with a hierarchy of items and directories much the same as a file system.

Figure 118 and Figure 119 on page 150 show a Gopher client interface.

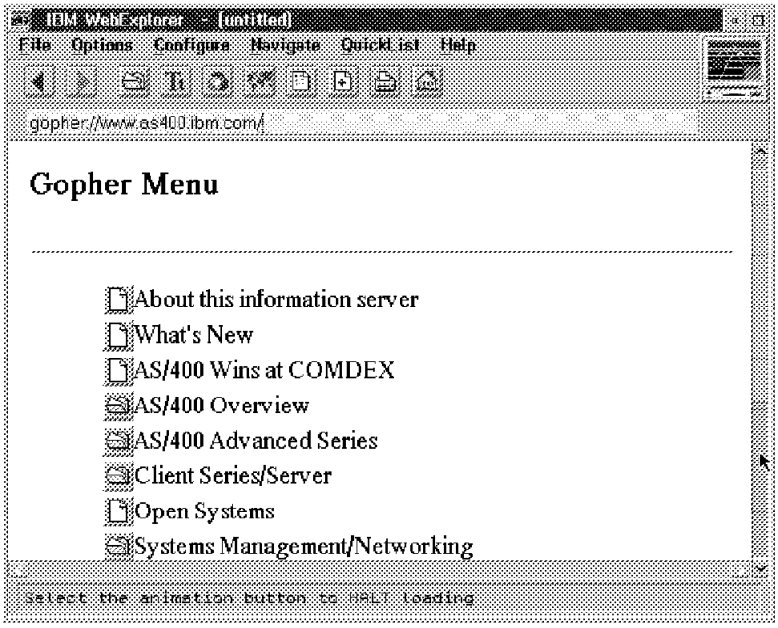


Figure 118. AS/400 Gopher Server View from an WebExplorer Client Display

```
                    AS/400 Gopher Menu List
                                                        System:   SYSNM001
   1=Select   4=Expansion   6=Print   9=Save

 Opt  Type  Item
  _   DOC   About this Gopher Server
  _   MENU  About Gopher, Mosaic, and the World-Wide Web
  _   MENU  Current OS/2 Gopher Client
  _   MENU  Almaden News, Weather, Network, and System Information
  _   MENU  IBMPC Keyword Search Server
  _   DOC   Lou Gerstner's 24 March 1994 Address to Securities Analysts
  _   MENU  Research Division Online Information
  _   DOC   Almaden RCF Support Staff (HELP STAFF)
  _   DOC   About Almaden INFO via Gopher
  _   MENU  The Internet at Your Desk
  _   MENU  Almaden INFO via Gopher
  _   MENU  INEWS - IBM News
  _   MENU  CMS Help
  _   MENU  Collected ANSWERS files from conference disks
  _   MENU  IBM Gopher Servers
  _   MENU  Search IBM phonebooks
                                                                  More ...
 F1=Help   F3=Exit   F7=Up   F8=Down   F12=Cancel
```

Figure 119. Gopher Client View From an AS/400 Display

With the advent of the World Wide Web in 1992 by CERN, the Gopher client is becoming less popular because most Web clients include a Gopher client. But, a Gopher server is still very popular because not all clients are web (or graphical) based.

The RFC1436 describes the Gopher protocol as follow:

"The Internet Gopher protocol and software follow a client-server model. This protocol assumes a reliable data stream; TCP is assumed. Gopher servers should listen on port 70 (port 70 assigned to Internet Gopher by the Internet Assigned Numbers Authority, IANA). Documents reside on many autonomous servers on the Internet. Users run client software on their desktop systems, connecting to a server and sending the server a selector (a line of text, which may be empty) through a TCP connection at a well-known port. The server responds with a block of text terminated by a period on a line by itself and closes the connection. No state is retained by the server."

This chapter talks about the Gopher on the AS/400 system and covers the following topics:

- The AS/400 Gopher Server: see 7.1, "The AS/400 Gopher Server" on page 151 for more information.
- The AS/400 Gopher Client: see 7.2, "AS/400 Gopher Client" on page 152 for more information.
- Gopher Search Engines: see 7.3, "Gopher Search Engines" on page 156 for more information.
- IBM Gopher Service Offering: see 7.4, "IBM Gopher Service Offering" on page 157 for more information.

7.1 The AS/400 Gopher Server

The AS/400 Gopher server is responsible for retrieving data from the AS/400 system and sending it to the client system. The server menus can retrieve data from the following sources:

- **Shared Folders:** The AS/400 Gopher server can send menus and files generated directly from shared folders very similarly to others platforms. A folder is sent as a menu, a document is sent as a document, and an ASCII PC file is sent as an IBM PC file to the Gopher client.
- **User spaces:** The AS/400 Gopher server can send user spaces that have a specific heading. The format of the heading is:

  ```
  Bin(31)   length of data in space
  Char(60)  reserved
  Char(*)   data
  ```

- **Database files:** The AS/400 Gopher server can send database and DDM files. The database file is sent to the Gopher client as a menu that displays the member as menu line items. The members of a file are sent as documents to the Gopher client. The actual data in the member is sent without converting fields that cannot be displayed.

  DDM files are used as references to the remote system where the data is actually stored. The Gopher client does not know that it is a DDM file.
- **Save files:** The AS/400 Gopher server can send save files as binary files that are not translated by the server or the client.

7.1.1 Items Supported by Gopher Server

Table 5 (Page 1 of 2). Supported Gopher Item Types

Type	Description	AS/400 Support
0	Text file (DOC)	Select to display the file ASCII PC file in a folder Non-ASCII documents in a folder Source physical file member Record of database file Database file member Distributed database(DDM) file member User space
1	Menu (MENU)	Folder Source physical file Database file Distributed database (DDM) file Library
5	IBM PC DOS file (DOS)	ASCII PC file in a folder
7	Search (SEARCH)	Source physical file Database file
8	Telnet (TELNET)	

Table 5 (Page 2 of 2). Supported Gopher Item Types		
Type	**Description**	**AS/400 Support**
9	Binary data (BINARY)	Save files ASCII PC file User space

7.2 AS/400 Gopher Client

With the Gopher client code on the AS/400 system, all of the users on PCs or non-intelligent workstations are able to access the Internet. The access using the Gopher code is menu-driven, allowing easy access to the Internet.

The next topics describe the following subjects in more detail:

- Setting up the Gopher Client
- Actions supported by Gopher client

7.2.1 Setting Up the Gopher Client

For setting up your Gopher client, there are two major commands that we cover in more detail:

- STRGPHR, which starts a Gopher client session.
- WRKGPHRCLT, where you can add new host names to access from the Gopher interface.

7.2.1.1 Start Gopher Client

To get started with Gopher Client, enter the STRGPHR command. This command starts the AS/400 Gopher client program and connects to this AS/400 system′s default server, which is an IBM server in Almaden.

Figure 120 shows the entry menu.

```
                          AS/400 Gopher Menu List
                                                          System:   SYSNM001
  1=Select   4=Expansion   6=Print   9=Save

Opt  Type    Item
 _   DOC     About this Gopher Server
 _   MENU    About Gopher, Mosaic, and the World-Wide Web
 _   MENU    Current OS/2 Gopher Client
 _   MENU    Almaden News, Weather, Network, and System Information
 _   MENU    IBMPC Keyword Search Server
 _   DOC     Lou Gerstner′s 24 March 1994 Address to Securities Analysts
 _   MENU    Research Division Online Information
 _   DOC     Almaden RCF Support Staff (HELP STAFF)
 _   DOC     About Almaden INFO via Gopher
 _   MENU    The Internet at Your Desk
 _   MENU    Almaden INFO via Gopher
 _   MENU    INEWS - IBM News
 _   MENU    CMS Help
 _   MENU    Collected ANSWERS files from conference disks
 _   MENU    Arctalk Answers (only of interest at Almaden)
 _   MENU    IBM Gopher Servers
                                                                    More...
F1=Help   F3=Exit   F7=Up   F8=Down   F12=Cancel
```

Figure 120. Gopher Menu List From an AS/400 Display

Option 1 is used to explore the various menus or read a document. If you choose a document item, you see the following display format:

```
                        AS/400 Gopher Document Browse
                                                                System:   SYSNM001
This is an experimental Gopher server, running Rice University's CMS Gopher
Server, release 2.3.3.  It is not a production server and may be withdrawn
at any time.  The primary use of this server is testing Gopher clients.

The Rice CMS Gopher client and server have been approved for use within
the Research Division under the Experimental Software program.

The Rice CMS Gopher client and server were written by Rick Troth of Rice
University.  I have added some tools to extract information from "typical"
IBM databases such as ANSWERS files, the Almaden INFO system, and the CallUp
phone books.

This server is maintained by David Singer, SINGER at ALMADEN.

                                                                        Bottom
F1=Help   F6=Print Document   F7=Up   F8=Down   F11=Save Document
F12=Cancel
```

Figure 121. Gopher View From an AS/400 Display

Also, you may want to see the name of the system where the information came from. Option 4 gives you this possibility, as follows:

```
                          AS/400 Gopher Menu List
                                                                System:   SYSNM001
  1=Select   4=Expansion   6=Print   9=Save

Opt  Type    Item
 4   DOC     About this Gopher Server
 _  .....................................................................
 _  :                      Expansion of item details:                     :
 _  :  Title  . :   About this Gopher Server                              :
 _  :                                                                    :
 _  :  Selector :   0/about.thisg                                        :
 _  :                                                                    :
 _  :                                                                    :
 _  :                                                                    :
 _  :                                                                    :
 _  :  Server . :   GOPHER-VM.ALMADEN.IBM.COM                            :
 _  :                                                                    :
 _  :                                                                    :
 _  :                                                                    :
 _  :                                                                    :
 _  :  Port . . :   70                                                   :
    :                                                                    :
F1= :                                                                    :
    :....................................................................:
```

Figure 122. Details of a Gopher List Entry

Other functionality available:

- Print using option 6, then use the WRKSPLF command to see the results.
- Save option, where you can save the item of interest in your library. This option does not support all item types. The save file (SAVF) should only be used if the file being saved is a save file. PC binary files should be saved to a document in a folder, and DOC item types should be saved to a source physical file such as QCLSRC in your library.

7.2.1.2 Work Gopher Client Host Name Table

The WRKGPHRCLT command allows you to maintain the table of servers defined in the host name file first member. This table contains information about servers for client access.

Figure 123 shows the list of servers that can be accessed and the entry display to add a new host.

```
                        Work with Gopher Client Entries
                                                              System:   SYSNM001
File . . . . . :   HOSTNAME        Member . . . . . :   HOSTNAME
  Library  . . . . . :     GOPHERLIB

  2=Change   4=Remove   5=Display

Opt  Host         Internet Name
 5   *DEFAULT     GOPHER.ALMADEN.IBM.COM
 _   AS400TEST    WWW.AS400.IBM.COM
 _   HOME         S403
 _   JUGHEAD      LIBERTY.UC.WLU.EDU
 _   MYSELF       SYSNM001.SYSNMAA.IBM.COM

                                                                        Bottom
===> ____________________________________________________________________________
F1=Help   F3=Exit   F6=Add   F7=Up   F8=Down   F12=Cancel
```

Figure 123. Work Gopher Client Entries Display

Option 5 allows you to see the information for the host.

```
                        Display Gopher Client Entry
                                                              System:   SYSNM001
File . . . . . :   HOSTNAME
  Library  . . . . . :     GOPHERLIB

Member . . . . . :   HOSTNAME

Host Name  . . . . . :   *DEFAULT

Port Number  . . . . :   70


Internet Name:
  GOPHER.ALMADEN.IBM.COM


F1=Help   F12=Cancel
```

Figure 124. Gopher Client Entry Display

The PF6 key allows you to create a new entry on this table. See Figure 125 on page 155

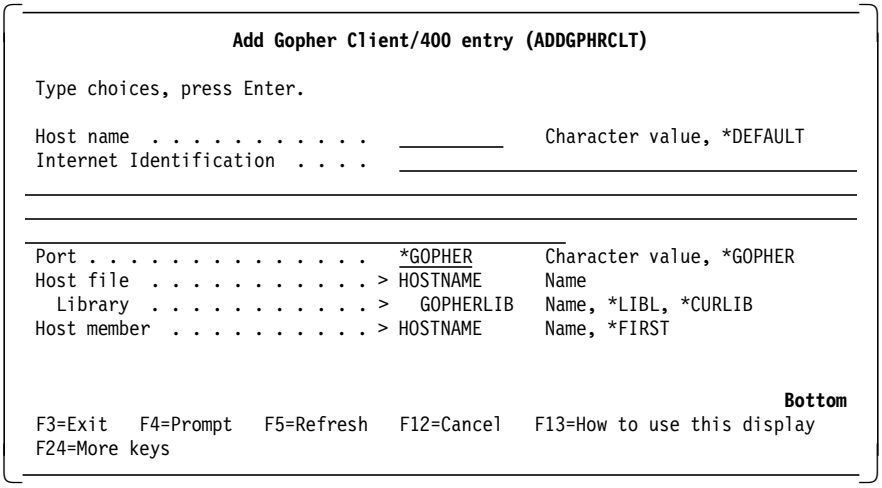


Figure 125. Add Gopher Client Entry Display

7.2.2 Actions Allowed For the Gopher Client User

Table 6. Supported Gopher Item Types

Type	Description	AS/400 Support
0	Text file (DOC)	Select to display the file. Print the file. Save the file.
1	Menu (MENU)	Select to display the items in the menu.
3	COS (PHONE)	No function supported.
4	Mac file (MAC)	Select to put in IBM PC DOS file. Save to put in IBM PC DOS file.
5	IBM PC DOS file (DOS)	Select to put in IBM PC DOS file. Save to put in IBM PC DOS file.
6	uuencoded file (UNIX)	Select to display. Print the file. Save the file.
7	Search (SEARCH)	Select to get search string.
8	Telnet (TELNET)	Supported for port 23 only.
9	Binary data (BINARY)	Select to save. Save file.
10	Graphic files (GIF)	Not Supported
11	Image data (IMAGE)	Select to put in IBM PC DOS file. Save to put in IBM PC DOS file.
12	TN3270 session (TELNET)	Supported for port 23 only.

7.3 Gopher Search Engines

On the Gopher space, we have some search engines available that are useful to get some information from the Internet.

We explain the Veronica and Archie engines in more details.

7.3.1.1 Veronica

Veronica is an index and retrieval system that can locate items on most of the Gopher servers in the Internet. The Veronica index contains about 10 million items from approximately 5500 Gopher servers (June 1994).

Veronica finds resources by searching for *words* in *titles*. It does not do a full-text search of the contents of the resources; it finds resources whose titles contain your specified search word (or words).

Veronica is used with a Gopher client. You choose *Veronica* from a menu of some Gopher servers, and enter a set of query words or special directives. When the search is finished, the results are presented as a normal Gopher menu. You may browse the discovered resources in this menu the same as you use any other Gopher menu.

Figure 126 shows how Veronica's search appears:

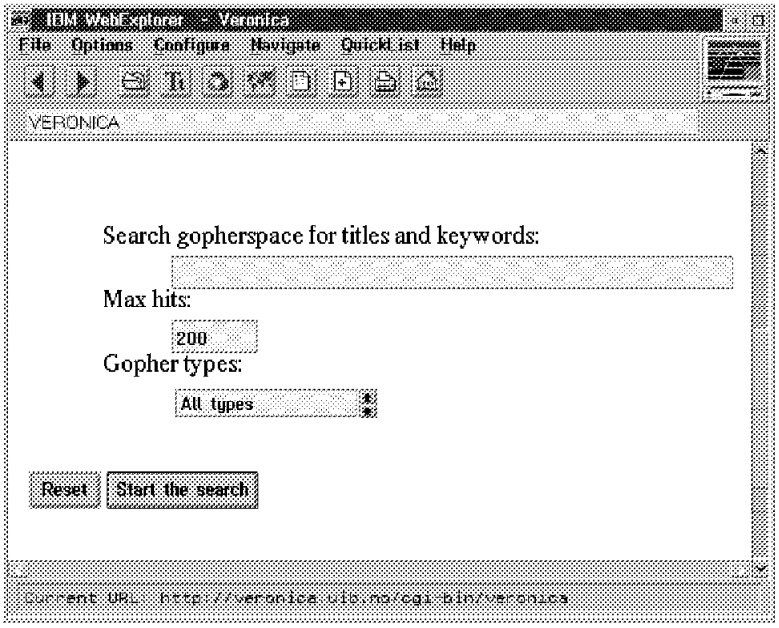


Figure 126. Veronica Search Display From a WebBrowser Viewer

7.3.1.2 Archie

Archie is a database of anonymous FTP sites and their contents. The software for it was written by the *Archie Group* (Peter Deutsch, Alan Emtage, Bill Heelan, and Mike Parker) at McGill University in Montreal, Canada, and they maintain the database as well.

Archie keeps track of the entire contents of a very large number of anonymous FTP sites, and allows you to search for files on those sites using various different kinds of filename searches.

Figure 127 on page 157 shows how an Archie's search appears:

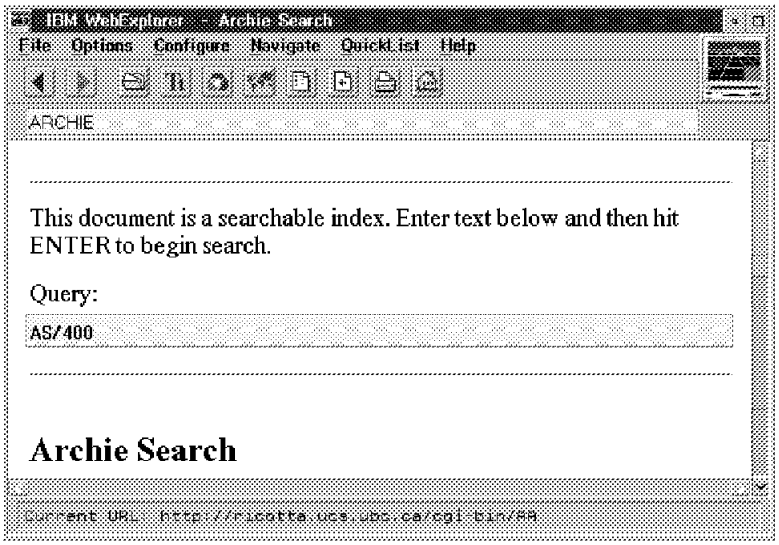


Figure 127. Archie Search Display From a WebBrowser Viewer

7.3.2 Other Gopher Sites

The following list contains some interesting Gopher sites:

- The *gopher://www.as400.ibm.com* from a Web/Gopher client.
- The *gopher://darban.cc.usm.edu/11/world* from a Web/Gopher client.
- The *http://gopher.usm.edu:70/1/world* from a Web client.
- The *http://vm.cfsan.fda.gov/vmgopher.html* from a Web client.

7.4 IBM Gopher Service Offering

AS400 Network Centric Computing Consulting:

IBM AS/400 Availability Services offers a complete set of startup, installation, and consulting services to integrate your company into network centric computing quickly. IBM offers the following services to complement OS/400:

On the AS/400 Gopher Internet Installation service, IBM configures TCP/IP to connect your system to the Internet, installs the Gopher client and server code, creates customized menus, and trains your operations and support staff. This service has been used to connect customers to the Internet since it was announced on May 9, 1995.

Chapter 8. Serving the World Wide Web from Your AS/400 System

The World Wide Web is one of the latest client-server based Internet services. It was developed by CERN (the European Laboratory for Particle Physics) in the late 1980s, when they were trying to find a way to allow widely dispersed research groups to easily access and display documents or view images that were stored on a server anywhere on the network. The result was a standard format for the documents, *hypertext*, that enabled them to be easily displayed by most types of display devices and also allowed links to other documents, *hyperlinks*, to be placed within documents. Once this service was made public as the World Wide Web in 1992, it became tremendously popular.

With the announcement of Internet Connection for AS/400, IBM offers AS/400 customers the capability to take an active part in the ever-growing and exciting World Wide Web. In doing so, IBM followed its commitment to Network Computing on the AS/400 system which means that everyone can access and distribute information, applications, and services provided by the network.

In the following chapter, we take a closer look at those features of Internet Connection for AS/400 that enable the AS/400 system to take part in the World Wide Web.

Important note

If you would like to follow along with the examples in this and other chapters you first will want to install the ITSO Company demonstration and other web based applications from the CD-ROM that came with this redbook. Please see Appendix A, “Installing the ITSO Company Demo” on page 285 for instructions on how to get your AS/400 up and running right away.

Important note

Install PTFs SF32078 (for 5763-TC1) and SF31077 (for 5763-SS1) before using the HTTP server for the AS/400 system.

8.1 HTTP Server

In this section, we are going to talk about HTTP, the Internet Protocol, and the different features that make an HTTP server a good Web server.

8.1.1 What is HTTP?

HyperText Transport Protocol (HTTP) is a fairly simple communication protocol used to transfer data, usually WWW pages between Web servers and Web clients. Web pages are written in a simple script-type language called HyperText Markup Language (HTML) so you can say that the role of HTTP is to transport HTML. Both client and server must support HTTP in order to send and receive HTML documents and, therefore, take part on the Web.

A Web client uses a Uniform Resource Locator (URL) to make requests of a Web server. A URL is a string of identifiers and instructions that point to a specific server and tell that server what object to process and how to process it.

There are different versions of HTTP, the one currently available is the fifth release of HTTP/1.0, which is anticipated to become an Internet "Best Current Practice" (BCP) specification. You can find more information about HTTP at URL:

`http://www.w3.org/pub/WWW/Protocols/`

8.1.2 How Does an HTTP Server Work?

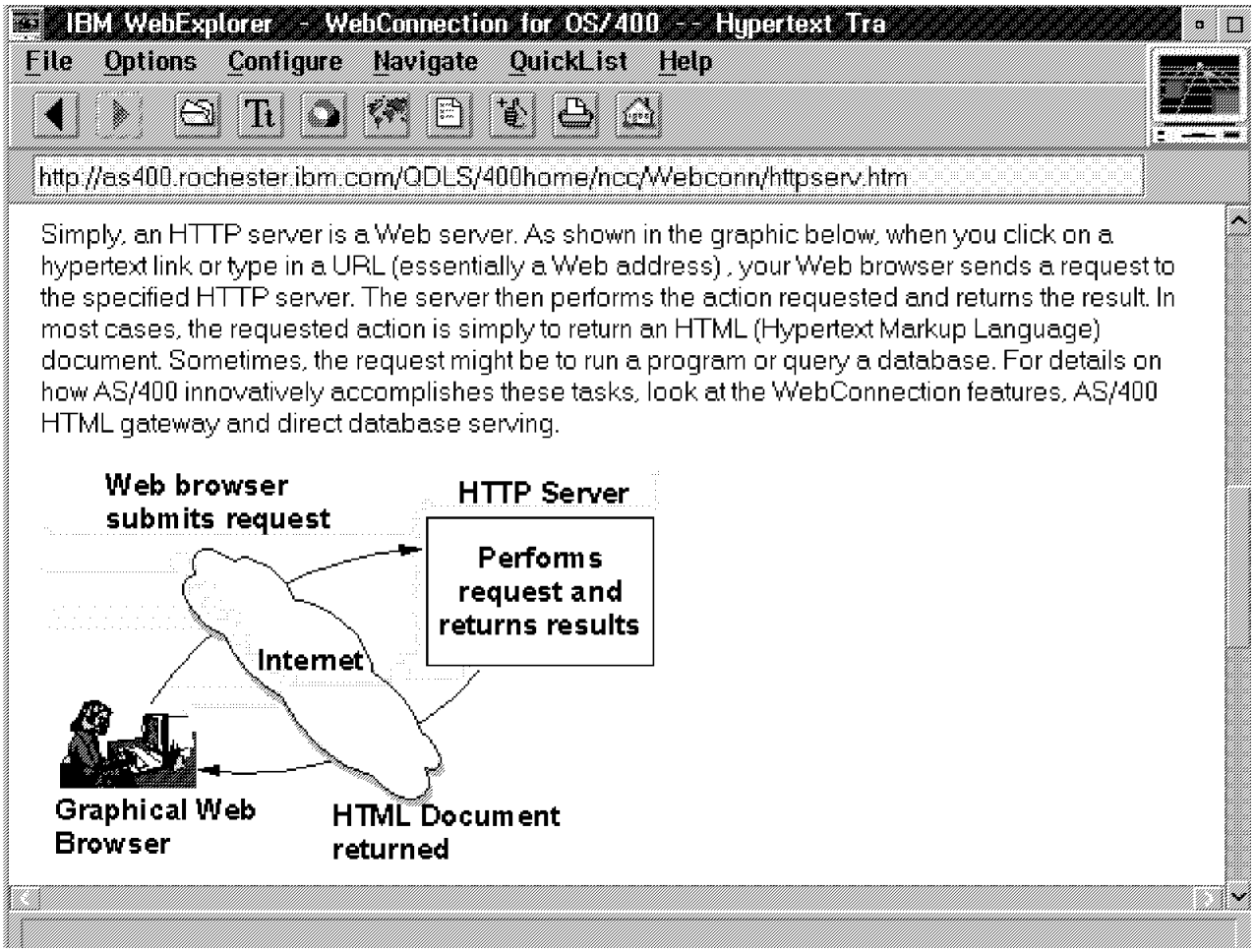


Figure 128. HTTP Server (note - the above URL is not valid anymore)

There are four basic steps in the HTTP protocol:

1. Connection
2. Request
3. Response
4. Close

During the connection, the Web browser (for example, WebExplorer, Mosaic, or Netscape) attempts to connect to the server (usually using port 80). If the browser cannot establish a connection to the server in a pre-configured amount of time, the browser displays an error message and aborts the user's request.

Once the connection to the HTTP server has been established, the browser sends a request for the desired resource to the server. If the server finds the resource, it sends it back to the browser and closes the connection. Otherwise,

an error message is sent back to the browser indicating that the requested resource is unavailable and the connection is closed.

If the server returned the requested resource, the browser then loads the returned resource and displays it or launches an external viewer to process multimedia resources, such as video or sound. If the server returned an error as a response to the resource request, the browser displays the returned error to the user.

Figure 128 on page 160 illustrates the procedure. When you enter a URL or click on a hypertext link, your Web browser sends a request to the Web Server. The server finds the requested resource and returns the result so that the browser can then display the resource. The Web browser used in Figure 128 on page 160 is the OS/2 WebExplorer.

8.1.3 HTTP Server Functions

The most basic role of an HTTP server is to provide browser clients access to HTML documents. However, there is further functionality, such as forms and clickable maps that are used to enhance the readability and usability of the document.

HTML document serving
This is the most basic feature of any Web server and the most widely used. Most Web pages today are static HTML documents stored on a server and then requested, interpreted, and displayed by a browser. See 8.4, “Hypertext Markup Language (HTML)” on page 184 for more details.

Multimedia serving
To simplify explanations or to liven up Web pages, HTML documents often include in-line images, or links to audio or video clips. There is nothing to prevent the AS/400 system from serving all types of Multimedia binaries from any of its stream file-based file systems, such as Root and QOpenSys. Most of the popular file type suffixes are built into the HTTP server. New ones can be added with the *AddType* directive. See the *TCP/IP Configuration and Reference* for a detailed explanation.

Logging For each request, the server logs the date, time, IP number of the client machine, and the text of the request. You can use this as a means of soliciting feedback for your company’s marketing organization on the most accessed portions of your company’s World Wide Web offerings. Refer to 8.4.3, “Logging the Access of the Web server” on page 188 for more details.

CGI-bin application support
CGI-bin support allows you to build interactive forms for use on the Web. You can write a program in C, RPG, or COBOL that can accept and interpret input from the form, perform an action, and return the resultant output to the client in HTML. CGI-bin is discussed in more detail in 9.1, “Common Gateway Interface Programs” on page 191.

Clickable maps (Image maps)
With clickable maps, users can click on different portions of an image and receive a different result. We talk more about clickable maps in 9.4, “Server Side Image map Support” on page 260.

DB2 World Wide Web

With DB2 World Wide Web, data from DB2 databases can be retrieved, formatted without application programming, and presented to users as information in the form of a Web page. Beyond just presenting information, DB2 World Wide Web can also be used as a front end to perform other database functions, such as updates and retrievals. Chapter 9.2, “DB2 World Wide Web Connection” on page 216 discusses this in more detail.

Workstation Gateway Server

Workstation Gateway Server converts AS/400 5250 displays into HTML and sends them to Web clients. It does this using its own built-in HTTP server code and not that used by the other Web applications.

8.2 Internet Connection for AS/400

Many of the features of Internet Connection for AS/400 are reliant upon the HTTP server. In this section, we look in detail at the HTTP server, its configuration, and its operation.

Note!

The HTTP server configuration **must** be modified by the user before the AS/400 system can be used as a Web server. By default, access to any Web page is forbidden.

If the documents are served from the QDLS file system (folders), user profile QTMHHTTP **must** have a directory entry added using the ADDDIRE command. Likewise, user profile QTMHHTTP **must** have object authority to all of the objects that are served.

8.2.1 HTTP Server Configuration Commands

The following list contains TCP/IP configuration commands for the TCP/IP HTTP Server:

1. CFGTCPHTTP (Configure TCP/IP HTTP)

   This menu of options is used to configure the HTTP server.

2. CHGHTTPA (Change HTTP attributes)

   This command is used to change the HTTP server attributes, such as the number of servers to start, whether to autostart the server, and so on. It is also an option on the Configure TCP/IP HTTP menu.

3. WRKHTTPCFG (Work with HTTP configuration)

   This command presents the user with a work-with display to allow HTTP administrators to add, change, display, or remove HTTP configuration options. The configuration format follows the CERN HTTP server configuration model and may look unfamiliar to many AS/400 users, but a traditional AS/400 work-with list display is used to interact with the file. It is also an option on the Configure TCP/IP HTTP menu.

There are several methods of starting the HTTP server once you have finished configuring it:

- The STRTCP command starts the server if the HTTP server attribute *AUTOSTART* is set to *YES.
- The STRTCPSVR SERVER(*HTTP) command starts the server if TCP/IP is already started and the HTTP server is not.
- The STRTCPSVR SERVER(*HTTP) RESTART(*HTTP) command restarts the HTTP server if both TCP/IP and the HTTP server are already started. This is useful when the configuration is changed and the HTTP server must be restarted in order for those changes to take effect. Since HTTP is basically a stateless protocol, this can be done while clients are accessing the AS/400 server. A client may just experience a slight delay while the server restarts.

The CFGTCPHTTP Display: The Configure TCP/IP HTTP display appears if the CFGTCPHTTP command is entered from the command line, or if CFGTCPAPP option 14 is selected.

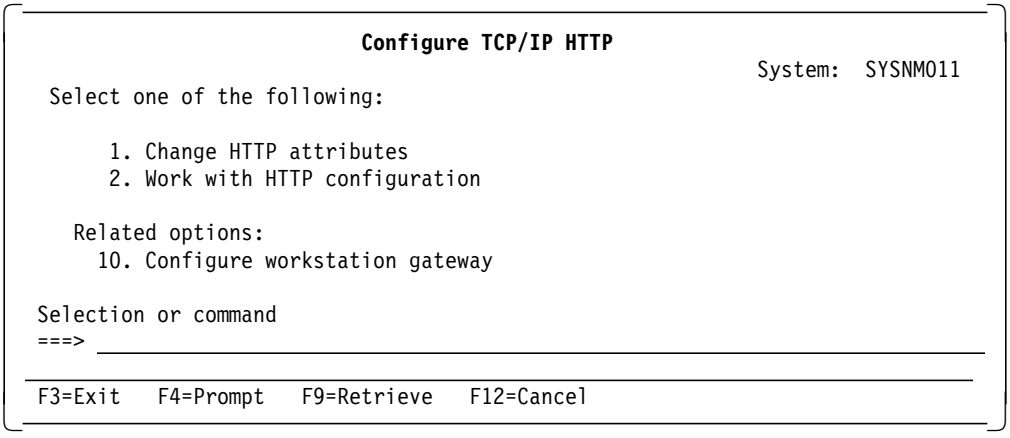


Figure 129. CFGTCPHTTP Display

- Option 1 - Prompts the CHGHTTPA command.
- Option 2 - Calls the WRKHTTPCFG command.
- Option 10- Calls the CFGTCPWSG command.

The CHGHTTPA Display: The Change HTTP Attributes display appears if the CHGHTTPA command is prompted from the command line or if CFGTCPHTTP option 1 is selected.

```
                Change HTTP Attributes (CHGHTTPA)
                                                        System:  SYSNM011
Type choices, press Enter.

Autostart  . . . . . .  . . .   *NO           *NO, *YES, *SAME
Number of server jobs:
  Minimum  . . . . . .  . . .   3            1-200, *SAME, *DFT
  Maximum  . . . . . .  . . .   5            1-200, *SAME, *NOMAX, *DFT
Coded character set identifier    00819       1-65533, *SAME, *DFT
Server mapping tables:
  Outgoing EBCDIC/ASCII table  .  *CCSID      Name, *SAME, *CCSID, *DFT
    Library  . . . . . .  . . .    _________  Name, *LIBL, *CURLIB *DFT

  Incoming ASCII/EBCDIC table  .  *CCSID      Name, *SAME, *CCSID, *DFT
    Library  . . . . . .  . . .    _________  Name, *LIBL, *CURLIB *DFT

                                                                    Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 130. Change HTTP Attributes Display

The values that are used on the preceding display are the default values.

Let's take time to look at some of the parameters.

Autostart (AUTOSTART)

Controls when the AS/400 system starts the HTTP server. Select *YES if you want the server to start whenever TCP/IP is started. If you use the STRTCPSVR command, this parameter is ignored and the HTTP server is started.

Number of Server Jobs (NBRSVR)

Use this parameter to specify the minimum number of HTTP server jobs to start when the HTTP server is started and the maximum number of HTTP server jobs to be used. If you set the maximum number of servers too high, this affects the performance of HTTP servers satisfying browser requests. You may want to increase this, however, if the HTTP servers are serving large documents that "tie-up" a server job for a long period.

Even if, for example, all of the HTTP servers are busy at a given moment, the new client requests coming into the HTTP server are queued. As soon as an HTTP server becomes available, the request is processed.

The remaining parameters are concerned with character translation and are similar to the parameters in Telnet attributes and FTP attributes. They are outside of the current discussion. For a complete explanation, see the *TCP/IP Configuration and Reference*.

8.2.1.1 The WRKHTTPCFG Display

This display appears if CFGTCPHTTP option 2 is selected or if the WRKHTTPCFG command is entered from a command line.

```
                      Work with HTTP Configuration
                                                          System: .SYSNM008
 Type options, press Enter.
   1=Add  2=Change  3=Copy  4=Remove  5=Display  13=Insert

      Sequence
Opt  Number  Entry

      00010   # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * >
      00020   # HTTP DEFAULT CONFIGURATION                                  >
      00030   # * * * * * * * * * * * * * * * * * * * * * * * * * * * * * >
      00040   #                                                            >
      00050   #                                                            >
      00060   #  HostName                    your.full.host.name           >
      00070   #                                                            >
      00080   #  The default port for HTTP is 80;  Should specify por      >
      00090   #  if port 80 is not used.                                   >
      00100   #  Port                         80                           >
      00110   #                                                            >
      00120   #  Enable                       GET                          >
                                                                    More...
 F3=Exit  F5=Refresh  F6=Print List  F12=Cancel  F17=Top  F18=Bottom
 F19=Edit Sequence
```

Figure 131. WRKHTTPCFG Display

Note: The '#' character is used to comment out a line.

```
                      Work with HTTP Configuration
                                                          System: .SYSNM008
 Type options, press Enter.
   1=Add  2=Change  3=Copy  4=Remove  5=Display  13=Insert

      Sequence
Opt  Number  Entry

      00130   #  Enable                       HEAD                         >
      00140   #  Disable                      {all others}                 >
      00150   #                                                            >
      00160   #  Map   /test/*  /as400/*                                   >
      00170   #  Pass  /as400SYS/* /QSYS.LIB/HTTPDEV.LIB/HTML.FILE/*       >
      00180   #  Pass  /as400/*    /QDLS/400HOME/*                         >
      00190   #  Pass  /QSYS.LIB/AS400LIB.LIB/HTML.FILE/*                  >
      00200   #  Pass  /QDLS/graphics/*                                    >
      00210   #  Fail  /QDLS/TESTING/*                                     >
      00220   #  Redirect  /some_url/*  http://some_server/some_url/*       >
      00230   #                                                            >
      00240   #  Search            <search_script_pathname>                 >
                                                                    More...
 F3=Exit  F5=Refresh  F6=Print List  F12=Cancel  F17=Top  F18=Bottom
 F19=Edit Sequence
```

Figure 132. More of the WRKHTTPCFG Display

When the HTTP server is started, this configuration information that is stored in file QUSRSYS/QATMHTTPC.CONFIG is used to process incoming client requests.

This configuration file can be saved and restored so before you make changes to a working HTTP server configuration, it may be worth backing it up.

You may find it easier to use the Source Entry Utility (SEU) to make lots of changes to the HTTP configuration file. If this is the case, then you should copy the HTTP configuration file member to a Source Physical File, modify it using SEU, then copy the changed file member back to QUSRSYS/QATMHTTPC. Time-wise, this is only worth doing if you are making major modifications to the file or do not feel comfortable with the HTTP server configuration interface. This method has one limitation. SEU can only handle files with the record length of 240 bytes which leaves 228 bytes for the actual data. When using SEU, you might get into difficulties with long path names in the directives.

An alternative way to edit the configuration file is to download the file to a PC using FTP and after editing it, upload it using FTP. Remember, though, that if you use any of these alternative ways to change the HTTP configuration, you should syntax check the file using the WRKHTTPCFG command.

Note: Before we discuss the configuration in detail, there are some considerations when working with HTTP configuration:

- Options 1, 2, 4 and 13 are only available if the user has *IOSYSCFG special authority, otherwise, only option 5 for display is allowed.
- Options 1, 2, 4 and 13 are only available to one user at a time. If the file is being modified by one user, only option 5 for display is available to subsequent users.
- ″Entry″ length is 640 characters, 55 of which are shown from the ″Work with HTTP Configuration″ display.
- Sequence numbers are reset to increments of 10 once configuration changes have been completed and processed.
- The configuration file is read sequentially.
- A valid entry comprises two parts, a directive and a value. The syntax should follow rules as defined by the ″CERN″ validation module.
- After configuration changes have been made, the file is checked by the CERN validation module and any invalid entries are highlighted.
- After modifying the HTTP Server configuration, the HTTP Server should be restarted. This can be accomplished without ending the server using the STRTCPSVR SERVER(*HTTP) RESTART(*HTTP) command.

8.2.2 HTTP Server Directives

AS/400 HTTP server directives determine how the HTTP server behaves. These directives are derived from the CERN configuration model.

The directives supported for the AS/400 HTTP server include:

- General settings:
  - HostName
  - Port
  - Enable
  - Disable
- Mapping rules (also known as URL translation rules):
  - Map
  - Pass

  - Fail
  - Redirect
  - Exec
- Filename suffix definitions (source type definitions for QSYS.LIB):
  - AddType
  - AddEncoding
  - AddLanguage
  - SuffixCaseSense
- Accessory programs (also known as accessory scripts):
  - Search
  - POST-Script
- Directory listings:
  - DirAccess
  - Welcome
  - AlwaysWelcome
  - DirReadme
  - DirShowDate
  - DirShowSize
  - DirShowBytes
  - DirShowOwner
  - DirShowDescription
  - DirShowMaxDescrLength
- Logging:
  - AccessLog
  - ErrorLog
  - LogFormat
  - LogTime
  - NoLog
- Time-outs:
  - InputTimeOut
  - OutputTimeOut
  - ScriptTimeOut

Note!

The minimum directives that you need to serve a document or execute a CGI are a *Pass* or *Exec* directive.

Let us look in detail at the HTTP server directives and their use.

8.2.3 General Settings

Hostname *full.server.host.name*

The HostName directive is used to override the host name value set using the ″Change local domain and host names″ option of the AS/400 Configure TCP/IP (CFGTCP) command. The directive only applies to those circumstances where HTTP server generates URLs that are self referential. We leave this directive commented out (#).

Port *port-no*

The *Port* directive is used to override the listening port value specified in the TCP/IP Services table for service *www* (accessible through the Configure TCP/IP (CFGTCP) ″Configure related tables″ option and then the ″Work with service table entries″ option). It may be useful to run the HTTP server on a non-standard port for testing purposes but without the need to change the Services Table. It is recommended that you use a port number greater than 1024 if you override the default HTTP sever port 80. To access the server with a HTTP server port of 1025, the required URL looks similar to this: `http://ServerName:1025/welcome.htm`

Enable & Disable *method-name*

The *Enable* and *Disable* directives cause the HTTP server to accept or reject incoming request URIs based on the *Method* coded in the URI. These *Methods* are described in the HTTP/1.0 specification. The AS/400 HTTP server supports *GET, HEAD,* and *POST*. All other methods are rejected.

In our HTTP server configuration, we changed the default values so that GET and POST were both enabled as a simple browser request uses the GET Method and many FORMS use the POST Method. If a Method is disabled, the HTTP server sends an error message to the Web browser informing the end user that a particular Method is not enabled.

Figure 133. Method GET Disabled on Server

Note: The error message is sent by the AS/400 HTTP server in the form of an HTML document. The error message, therefore, is the same regardless of the type of Web browser.

Defaults

Enable GET

Enable HEAD

Disable POST

If the *methods* are commented out, then the defaults values determine what is enabled or disabled. Therefore, there should be no value in adding a directive such as:

```
Enable GET
```

8.2.4 Mapping Rules

Mapping rules (which are also known as URL translation rules) allow the Web administrator to define a virtual hierarchy of Web resources that are presented by the HTTP server. This virtual hierarchy is independent of the physical file system (or systems) on which the resources are actually stored. This is very useful since:

1. It allows resources to be physically relocated without changing the apparent hierarchy and its implied associations.
2. The details of the underlying physical file systems can be hidden from client applications accessing the HTTP server.

The second reason is especially important for the HTTP server because of the differences between some of the AS/400 file systems and the tree structured hierarchical UNIX file systems upon which the HTTP protocol and the URL scheme are based.

8.2.5 URL Mapping Overview

The mapping directives *Map, Pass, Fail, Redirect* and *Exec* are used to specify a set of rules that define a process for translating and processing the path (and file) information contained in a URL that is received in a client request URL.

Each of the mapping directives may be specified multiple times and in any combination, and may be placed anywhere in the HTTP server configuration. When the HTTP server configuration is read during initialization, the mapping directives are stored, in order of appearance, in an internal mapping rule table.

The named object in each incoming URL is compared against each of the mapping directives in turn. If a directive applies, the specified translation or action is carried out. Depending on the directive that was matched, rule processing is either suspended or continues with the next rule using the newly translated URL.

8.2.6 Mapping Directive Descriptions and Examples

Note: It is important to note that the following directives are case-sensitive. A *template* checks the input value for a match letter-by-letter. This has important security implications, especially for the *Fail* directive that we cover in that section.

Map *template replacement*

If the URL matches the *template*, substitute the *replacement* string for the *template* and continue to the next rule using the resulting URL.

Example

Your Welcome.htm document is reached through the following path in the QDLS file system:

```
http://ServerName/QDLS/FOLDER1/WEBSTUFF/HTML/Welcome.htm
```

This is quite a complex URL for a client to remember and type so it makes sense to present a simpler, virtual structure to the client. Using the *Map* directive, we can modify the HTTP server configuration as follows:

```
Map /info/*  /QDLS/FOLDER1/WEBSTUFF/HTML/*
```

Now the client can type a URL such as:

```
http://ServerName/info/Welcome.htm
```

The Map statement replaces the /info/* value with the /QDLS/FOLDER1/WEBSTUFF/HTML/* replacement string and the HTTP server continues with further rule processing using the resulting URL.

Pass *template {replacement}*

If the URL matches the *template*, rule processing is terminated.

If the optional *replacement* string is present, substitute it for the *template* string and complete the client request using the resultant URL.

Example

We want to allow Web clients to browse the document Welcome.htm; therefore, we **must** code a *Pass* statement:

```
Pass /info/Welcome.htm  /QDLS/FOLDER1/WEBSTUFF/HTML/Welcome.htm
```

When a Web client requests URL:

```
http://ServerName/info/Welcome.htm
```

The HTTP server *Passes* (or allows) the request. In this case, however, we used the optional *replacement* string to modify the request through the real file structure to the Welcome.htm document.

Note: A generic statement such as Pass /* is a simple way of allowing access to any document, file member, and so on, but the security implications are such that we recommend using specific *Pass* statements to only those objects that you want the public to see.

Fail *template*

If the URL matches the *template* rule, processing is terminated and access is refused. An error code is returned to the client.

Figure 134. A Failed URL Request

Because these *templates* are case sensitive and some file systems are not, it is easier to secure objects using the *Pass* directive than the *Fail*. This is because you need to add a statement for every possible combination of upper and lower case letters in the object you are trying to secure.

Example

You want to refuse access to object *secure.htm* so, considering case sensitivity, you add two *Fail* statements as shown in the following example:

```
Fail  /info/secure.htm

Fail  /info/SECURE.HTM
```

You assume that you are secure since both upper and lower case possibilities have been covered. It is, however, possible to access the document using a URL such as:

```
http://ServerName/info/SeCuRe.htm
```

Since the value `SeCuRe.htm` does not match either of your *Fail* statement *templates*, the request does not fail and the document is accessed.

Once again, we recommend using specific *Pass* statements and maybe, as an additional measure, a generic `Fail  /*` *"*catch-all*"* at the bottom of your configuration file, although it should not be necessary if your Pass statements are explicit enough as requests should fail by default.

Redirect *template replacement*

When documents, or entire virtual trees of documents, are moved from one server to another, the *Redirect* directive can be used to redirect the request to another server.

If the URL matches the *template*, rule processing is terminated. Substitute the *replacement* string for the *template* string and send a *Redirect* containing the resulting URL back to the requester.

Using this directive, the requester is transparently ″re-routed″ to the URL specified in the :hp1replacement string.

Example

```
Redirect  /info/*  http://AnotherServer/NewDir/Welcome.htm
```

This redirects a request for any document in the */info* directory to the *Welcome.htm* document on the host called *AnotherServer*.

Exec *template replacement*

A match with the *template* specified on an *Exec* directive indicates that the URL should be interpreted as referring a program to be executed by the HTTP sever on behalf of a client/browser.

On this directive, both *template* and *replacement* are required to end with a wildcard (asterisk *).

A *template* identifies the virtual path to a program (or you can think of this as a virtual directory that contains one or more of your programs). It is replaced with *replacement* which is the actual path to the program on this server.

Note: AS/400 executable programs are required to be program objects in a library in the QSYS.LIB file system. Therefore, the URL specified by *replacement* must be of the form `/QSYS.LIB{/lib.LIB}/*`

The beginning of the text represented by the wildcard is assumed to be the actual name of the program.

Example

```
Exec  /info/cgipgm/*  /QSYS.LIB/as400cgi.LIB/*
```

The *Exec* directive maps the `/info/cgipgm` directory into the `/QSYS.LIB/as400cgi.LIB` library where, in this example, our CGI programs are kept.

Note: You should be aware that the position of directive statements in the HTTP server configuration is important. For example, a statement such as `Pass  /*` positioned before a *Map* or *Exec* statement prevents rule processing from ever reaching them, as every incoming URL matches the generic *catch-all Pass* statement, and rule processing ends. Always ensure that any directive statements that you add do not adversely affect subsequent statements.

8.2.7 Filename Suffix Definitions

Filename Suffix definitions are used to define the meaning associated with the suffix part of the file name.

We are not talking about this area in this book. For a detailed explanation, see the *TCP/IP Configuration and Reference*.

8.2.8 Accessory Programs

The *Search* and *POST-script* directives are used to specify a ″default″ program to use when a CGI ″search″ or a POST request are received but do not match any *Exec* directives.

We are not talking about this area in this book. For a detailed explanation, see the *TCP/IP Configuration and Reference*.

8.2.9 Directory Listings

DirAccess On | Off | Selective

This directive enables/disables the generation of a directory listing document for URLs that do not refer to a specific IFS file or QSYS.LIB member.

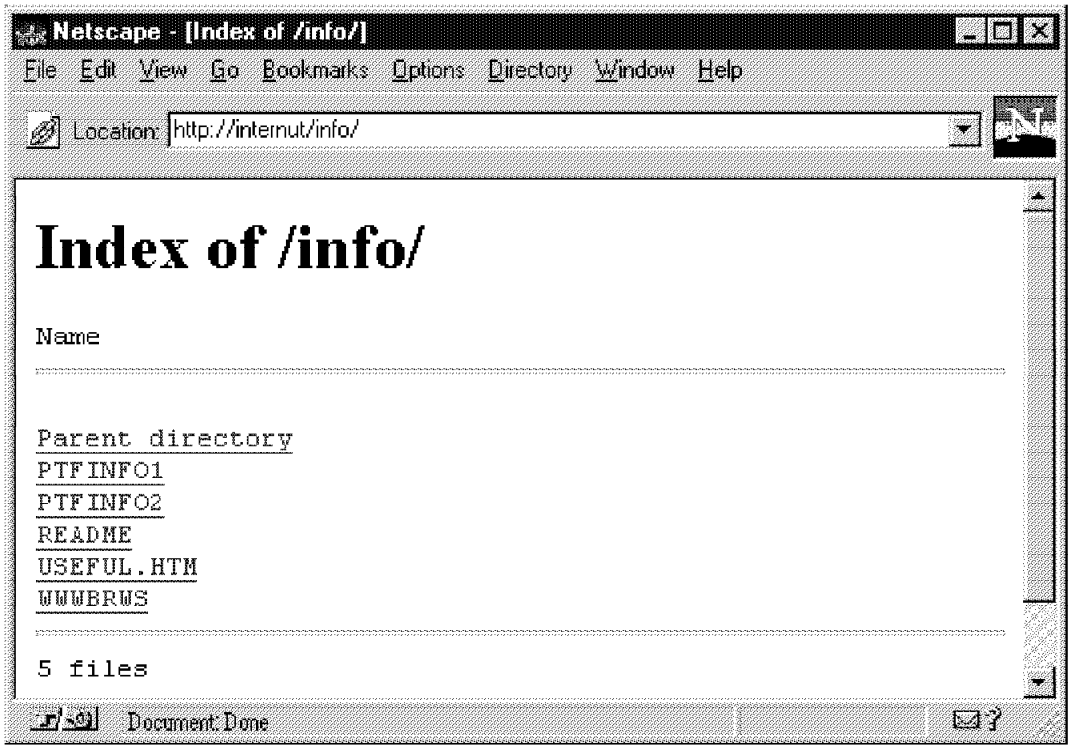


Figure 135. Typical Directory Listing Document

Note: The HTTP server's decision to generate a directory listing is also affected by the *Welcome* and *AlwaysWelcome* directives.

On Generation of directory listings documents is enabled. It is enabled for URLs of the form `/QSYS.LIB{/lib.LIB}/file.FILE`. Other QSYS.LIB forms are rejected.

Off Generation of directory listing documents is disabled for all URLs.

Selective Generate a directory listing document when an entity named **wwwbrws** is present.

This enables the system administrator to selectively allow directory listings of specific files (in

/QSYS.LIB{/lib.LIB}/file.FILE) and directories (IFS directories).

QSYS.LIB Create a file member named **wwwbrws** to allow the generation of a directory listing document of the file containing this member.

IFS Create a directory, a file object in the directory, a folder, or document within the folder called **wwwbrws** to generate a directory listing of the directory or folder containing the object **wwwbrws**. Figure 136 on page 176 shows folder LEETEST in QDLS that contains three HTML files and a file *wwwbrws* that allows the generation of a directory listing document for this folder.

Welcome *name*

The *Welcome* directive specifies the *name* of a file to be used when no file is specified on an incoming URL. It is a useful method of forcing a client to a particular document such as your *Home Page*.

When an incoming URL has no QSYS.LIB file, IFS file, or QDLS document specified in it, the HTTP server, depending upon the state of the *AlwaysWelcome* directive, scans the file or directory in the URL for a welcome file matching the *name (or names)* specified. If an object is found that matches a *name*, then it is returned to the client. If it does not find a matching object, then a directory listing document may be returned to the client in accordance with the setting of the *DirAccess* directive.

Table 7. HTTP Server Welcome Object Processing

Condition			**Result**
Welcome name exists	**AlwaysWelcome directive**	**URL has trailing '/'** 1	
Yes	ON (Default)	Ignored	Serve Welcome
Yes	OFF	Yes	Serve Welcome
Yes	OFF	No	Check DirAccess (see Table 8)
No	Ignored	Ignored	Check DirAccess (see Table 8)

1 Refers to integrated file system directory, AS/400 file or QDLS folder with no file specified. The URL must always point to a directory for welcome or directory processing.

Table 8 (Page 1 of 2). Directory Processing

Condition		**Result**
DirAccess directive	**'wwwbrws' in directory**	
OFF (Default)	Ignored	Error 403
ON	Ignored	Serve Directory

Table 8 (Page 2 of 2). Directory Processing		
Condition		**Result**
DirAccess directive	**'wwwbrws' in directory**	
Selective	Yes	Serve Directory and contents of README file if enabled with DirReadme
Selective	No	Error 403

It is possible to add multiple *Welcome* directives should you need a welcome page with a different name in a different file.

Note: Welcome processing is further controlled by the setting of the *AlwaysWelcome* directive.

AlwaysWelcome On | Off

The *AlwaysWelcome* directive can be used to further control welcome file processing for URLs that do not refer to a specific resource.

Off If the URL ends with a trailing slash (/), the welcome file (if one exists) is returned to the client.

If the URL does not end with a trailing slash, the URL is processed in accordance with the setting of the *DirAccess* directive.

On The welcome file (if one exists) is always returned regardless of the presence or absence of a trailing slash in the URL.

The remaining directives in this section control the appearance of directory listing documents generated by the HTTP server.
Figure 136 on page 176 shows a directory listing of folder LEETEST in QDLS with directive *DirReadme* set to *Top*, *DirShowDate, DirShowSize, DirShowBytes, DirShowOwner, DirShowDescription* set *On*, and *DirShowMaxDescrLength* set to *25* (the default).

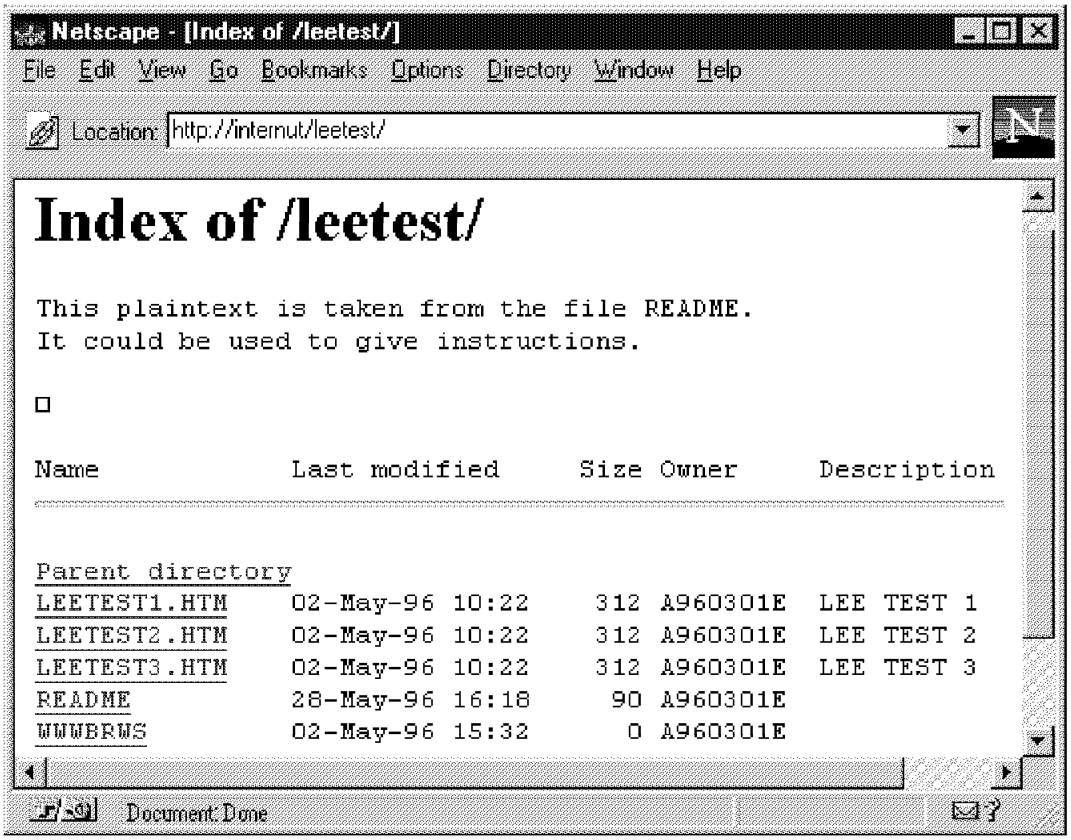


Figure 136. Directory Listing

See the following list for a brief explanation of these directives. The default setting for each directive is shown underlined.

DirReadme Top | Bottom | Off
: This directive controls if and where README information is placed in a directory listing document.

  If README is a member name in QSYS.LIB or a file name in the IFS, then the plain text it contains is displayed on the directory listing document.

DirShowDate On | Off
: Controls whether the modification date is displayed for each entity in the listing.

DirShowSize On | Off
: Controls whether the size is displayed for each entity in the listing.

DirShowBytes On **| Off**
: Controls whether, for objects smaller than one kilobyte, the exact size in bytes is displayed.

DirShowOwner On | Off
: Controls whether the owning USERID is displayed for each entity displayed in the listing.

DirShowDescription On **| Off**

Controls whether the descriptions of documents are included in the listing. For QSYS.LIB members, the member description is shown. For members that contain HTML and whose description is blank, or for HTML files in the IFS, the description is extracted, if possible, from the TITLE section of the document.

DirShowMaxDescrLength *len*

Sets the maximum number of characters allowed for description text.

The order in which the resultant values of these directives are displayed on the listing document is fixed regardless of the order the directives are entered in the HTTP server config.

8.2.10 Logging

AccessLog *log-file-name {max-size}*

The *AccessLog* directive specifies the *log-file-name* of the access log file. When the HTTP server is started, a member is automatically created in QUSRSYS/*log-file-name* with a member name based on time and date.

The optional *max-size* parameter specifies the maximum size of the file in kilobytes. If 0 is specified, the log file keeps growing until STRTCPSVR SERVER(*HTTP) RESTART(*YES) is executed. The default value is 2 megabytes.

If this directive is not explicitly specified in the HTTP server configuration, no access logging is performed.

```
                          Display Physical File Member
File . . . . . :   LEEACLO3           Library  . . . . :   QUSRSYS
Member . . . . . :   Q0960507         Record . . . . . :   1
Control  . . . . .   W60              Column . . . . . :   1
Find . . . . . .   ____________________________________
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+...
[07/May/1996:12:38:34 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/qpgmsrc.file/
[07/May/1996:12:38:51 +0000] w3proxy-b.SYSMA99.ibm.com GET /QSYS.LIB/A960301C.
[07/May/1996:12:39:33 +0000] w3proxy-b.SYSMA99.ibm.com GET /QSYS.LIB/A960301C.
[07/May/1996:12:39:53 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/QPGMSRC.FILE/
[07/May/1996:12:40:32 +0000] w3proxy-b.SYSMA99.ibm.com POST /QSYS.LIB/A960301C
[07/May/1996:12:41:01 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/WEBCGR00.PGM
[07/May/1996:12:41:10 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/QPGMSRC.FILE/
[07/May/1996:12:41:16 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/QPGMSRC.FILE/
[07/May/1996:12:41:25 +0000] w3proxy-b.SYSMA99.ibm.com POST /QSYS.LIB/A960301C
[07/May/1996:12:46:55 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/QPGMSRC.FILE/
[07/May/1996:12:47:00 +0000] w3proxy-b.SYSMA99.ibm.com GET /QSYS.LIB/A960301C.
[07/May/1996:12:47:05 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/qpgmsrc.file/
[07/May/1996:12:48:09 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/qpgmsrc.file/
[07/May/1996:12:48:16 +0000] w3proxy-b.SYSMA99.ibm.com GET /PGMS/QPGMSRC.FILE/
[07/May/1996:12:48:22 +0000] w3proxy-b.SYSMA99.ibm.com POST /QSYS.LIB/A960301C
                                                                        More...
F3=Exit   F12=Cancel   F19=Left   F20=Right   F24=More keys
```

Figure 137. Typical Access Log Member (Left)

```
                          Display Physical File Member
File . . . . . :   LEEACL03            Library  . . . . :   QUSRSYS
Member . . . . . :   Q0960507          Record . . . . . :   1
Control  . . . . .   W60               Column . . . . . :   60
Find . . . . . .   ____________________________________
6....+....7....+....8....+....9....+....0....+....1....+....2....+....3....+..
/PGMS/qpgmsrc.file/stdoutin.mbr HTTP/1.0
/QSYS.LIB/A96O3O1C.LIB/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
/QSYS.LIB/A96O3O1C.LIB/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
/PGMS/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
 /QSYS.LIB/A96O3O1C.LIB/WEBCGRO0.PGM HTTP/1.0
/PGMS/WEBCGRO0.PGM HTTP/1.0
/PGMS/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
/PGMS/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
 /QSYS.LIB/A96O3O1C.LIB/WEBCGRO0.PGM HTTP/1.0
/PGMS/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
/QSYS.LIB/A96O3O1C.LIB/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
/PGMS/qpgmsrc.file/stdoutin.mbr HTTP/1.0
/PGMS/qpgmsrc.file/stdoutin.mbr HTTP/1.0
/PGMS/QPGMSRC.FILE/RDSIN.MBR HTTP/1.0
 /QSYS.LIB/A96O3O1C.LIB/WEBCGRO0.PGM HTTP/1.0
                                                                      More...
F3=Exit   F12=Cancel   F19=Left   F2O=Right   F24=More keys
```

Figure 138. Typical Access Log Member (Right)

Access logging is a useful aid for analyzing who is accessing your system, how frequently your system is being accessed and which Web pages are being accessed. This is useful for both security auditing and marketing related analysis.

ErrorLog *file-name {max-size}*

The *ErrorLog* directive specifies the *file-name* of the access log file. When the HTTP server is started, a member is automatically created in QUSRSYS/*file-name* with a member name based on time and date.

The optional *max-size* parameter specifies the maximum size of the file in kilobytes. If 0 is specified, the log file keeps growing until STRTCPSVR SERVER(*HTTP) RESTART(*YES) is executed. The default value is 2 megabytes. If this directive is not explicitly specified in the HTTP server configuration, no access logging is performed.

LogFormat DDS **| Common**

The *LogFormat* directive defines which format the log files are saved in. *Common* file format is a source physical file with the total record length of 375 bytes. It is very hard to process using PDM because the SEU can handle only records up to 240 bytes long. *DDS* file format is divided into three fields and has a total record length of 375 bytes. We have included into library ITSOIC400 the object and source code for program PRTWWWLOG, which is a quite simple ILE RPG program that produces a print of both file formats.

LogTime LocalTime **| GMT**

Specifies how the log file entries are time stamped.

NoLog *template{ template{...}}*

The *NoLog* directive is used to disable the logging of HTTP server accesses for selected groups of client/browser host names and host addresses. It can be used to filter out log entries of your regular

users and leave just those entries that may be considered security breaches, or those of interest for marketing reasons.

Examples

```
NoLog  9.180.*.*

NoLog  *.local.users.com

NoLog  *.abc  *.pqr  *.xyz
```

The *template* parameter is either a dotted decimal host address or a dotted Internet Host name. An asterisk ('*') may be used as a wild card for any of the terms.

This directive is relevant only when the *AccessLog* directive is specified. It may be specified multiple times in the HTTP server config and more than one *template* may be specified on each *NoLog* directive.

8.2.11 Time Outs

These directives allow the HTTP server administrator to set time limits for the various parts of the request handling process. All three directives require a *time-spec* argument in one of the following forms:

```
2 minutes

10 minutes

1 hour
```

Whole minutes are the smallest increment that can be set.

InputTimeOut *time-spec*

The time to wait for a client to send the MIME-header part of the request (the message body, if there is one, is read during subsequent processing).

Default: The default value is 2 minutes.

OutputTimeOut *time-spec*

The time to allow for sending the response back to the client. This time may need to be increased if large files are being served over slow links.

Default: The default value is 20 minutes.

ScriptTimeOut *time-spec*

The time to wait for a CGI program to finish. If the program does not finish in the allotted time, processing of the entire request is forcefully terminated and an error code returned to the client.

Default: The default value is 5 minutes.

8.2.12 Updates to Existing TCP/IP Commands

The following is a list of TCP/IP commands that are updated to support the TCP/IP server.

1. CFGTCPAPP (Configure TCP/IP Applications)

   The menu of options is updated to include an option to call the CFGTCPHTTP command.

2. STRTCPSVR (Start TCP/IP Servers)

The SERVER parameter is updated to allow the special value *HTTP (Start HTTP server).

A new *RESTART parameter is added, which is only valid for a server value of *HTTP. This gives the HTTP server a restart option, which is common with CERN-like HTTP servers and gives the customers a greater degree of control.

All HTTP server jobs run in batch mode under the QSYSWRK subsystem, and are started with the STRTCPSVR SERVER(*HTTP) command and ended with the ENDTCPSVR SERVER(*HTTP) command.

8.2.13 URI Interpretation

The AS/400 HTTP server acts on browser requests in accordance with the content of the Request-URI (Universal Resource Identifier) from the request line.

The following description shows how URLs for each file system are written and handled.

Requests for QSYS.LIB objects

Physical Files and *Source Physical Files*

The format for a physical file/source physical file URL is:

```
QSYS.LIB/library.LIB/file.FILE/member.MBR
```

If a member is identified, the member is sent after being translated from EBCDIC to ASCII.

Note: The member type **must** be HTML for the contents to be identified as HTML, otherwise the content is identified as plain text.

If the file is specified with no member, a list of the members is sent as an HTML document using the member descriptions as identifiers. If a member HEADER is present, it is added as a description at the beginning of a list. If a member README is present, it is added as a description at the end of the list.

Database File: The handling of database files is discussed in the 9.2, “DB2 World Wide Web Connection” on page 216.

Requests for QDLS objects

The format for a QDLS URL is:

```
QDLS/folder1/folder.../document
```

PCFILE objects in QDLS are sent ″as-is″ with the file extension determining the type. A list of types is provided with the server, and others may be added.

Program call (CGI interface)

The format for a program call is:

```
QSYS.LIB/library.LIB/program.PGM
```

The client can request that the HTTP server run a program. The program identified by the client is expected to produce an HTML document that the server can serve back to the requesting client. The *Common Gateway Interface (CGI)* specification defines how the server is expected to call such external programs, and how those programs should return the document that they produce. Refer to 9.1, “Common Gateway Interface Programs” on page 191 for more details on CGI.

Note: The CGI program must be in the QSYS.LIB file system which is the only file system from which you can execute programs on the AS/400 system.

DB2 database requests

DB2/www is an extension to the AS/400 HTTP server. The HTTP server recognizes requests for DB2/www when parsing the URL of a client's request and passes such requests to the DB2/www server extension.

The format for a DB2/www URL is:

```
QSYS.LIB/library.LIB/db2www.PGM/{macro-file}/{command}
```

The request for DB2/www is a special case of a program call; the program to be called is *db2www*. The server recognizes a DB2/www request by the program name (db2www) and passes the request to the db2www extension of the server. Refer to 9.2, "DB2 World Wide Web Connection" on page 216 for more details.

Note: The DB2www program must be in the QSYS.LIB file system, the only file system from which you can execute programs on the AS/400 system.

MAPS Image map requests are another form of program call (CGI) requests.

The format for a image map request is:

```
QSYS.LIB/library.LIB/imagemap.PGM/{imagemap-spec}
```

HTML maps can be served by either a server-provided image mapping program (image map) or a user-provided image mapping program. Refer to 9.4, "Server Side Image map Support" on page 260 for more details about maps.

Note: The image map program must be in the QSYS.LIB file system which is the only file system from which you can execute programs on the AS/400 system.

Request for other file systems

Requests for objects from other file systems are processed using the Integrated File System.

- Root
- QOpenSys
- QLANSrv

Files from these file systems are considered to be binary files not requiring translation. They are sent as-is to the client.

8.3 HTTP Security Considerations

Security is a major consideration when putting any computer system on the Internet. Many of these considerations are covered in Chapter 10, "Security and Audit on the Internet" on page 267.

In this section, we briefly cover some security points specific to the HTTP server configuration.

8.3.1 Using Directives for Security and Access Control

The HTTP server administrator controls the behavior of the HTTP server. The HTTP server does *not* do anything that the server administrator has not explicitly configured it to do.

Several features of the HTTP server ensure that the administrator maintains this control:

- The default fail rule means that only requests that are authorized by the Web administrator are honored; other requests fail.
- Explicit CGI enablement means that no CGI programs run unless specifically authorized.
- Only CGI programs are run.
- Only the read HTTP methods, GET, POST, and HEAD are supported.

8.3.2 The Default Fail Rule

The HTTP server fails, by default, all incoming requests *unless* the URL, as translated by any preceding *map* directives, matches a *Pass, Redirect,* or *Exec* directive that has been explicitly coded by the HTTP server administrator.

- A match with a *Pass* directive statement enables the HTTP server to serve a document.
- A match with a *Redirect* directive statement causes the HTTP server to return a *Redirect Response* to the client application. No AS/400 data is accessed.
- A match with an *Exec* directive statement enables the HTTP server to execute a CGI program on behalf of the client.

The HTTP server configuration file, as shipped, does *not* contain any *Pass, Redirect,* or *Exec* statements.

8.3.3 Explicit CGI Enablement

The HTTP server does not execute a user-defined CGI program unless the HTTP server administrator has explicitly enabled it by coding an *Exec* directive. The HTTP server administrator can, for example, limit CGI requests to a specific library in QSYS.LIB.

Important!

It is the HTTP server administrator's responsibility to verify that any CGI program that is enabled does not violate the customer's security policies for the AS/400 system on which the HTTP server is running.

IBM recommends that the HTTP administrator move both the QTMHIMAG *PGM and the DB2WWW *PGM from the QTCP library to their own CGI library. This allows users to run CGI programs while limiting access to the QTCP library.

8.3.4 HTTP Server Runs Only CGI Program

To run properly, programs that are called by the HTTP server must conform to the HTTP server CGI interface. When the HTTP server is enabled to call a particular program on behalf of a remote HTTP client application, the program is called and the output is returned through the HTTP server CGI interface.

8.3.5 The HTTP Server is a Read-Only Server

The HTTP specification defines seven ″methods″ that represent seven types of requests that a remote HTTP client application can send to an HTTP server. The HTTP server implements only the GET, HEAD, and POST requests. The HTTP server does *not* implement two other methods, PUT and DELETE. This makes it impossible for a request from an external client to cause an AS/400 object to be overwritten.

8.3.6 Do Not Allow All Programs in QTCP.LIB to be Executed

The formal IBM manual (*OS/400 TCP/IP Configuration and Reference*) mentions that IBM recommends that the HTTP administrator move both the QTMHIMAG.PGM and DB2WWW.PGM to your own CGI library to avoid the security problem of allowing *all* programs in QTCP.LIB to potentially be executed (or called) by the HTTP server. This is prudent advice because no one is really sure what damage a vandal can cause by calling FTP through the HTTP server.

The problem is that after you have moved QTMHIMAG.PGM and DB2WWW.PGM to your own CGI library, then any PTF updates to your software is not automatically made because it is the original program in QTCP.LIB that has been updated.

Another (untested) solution might be to temporarily remap the incoming URL to a bogus one, Fail /QSYS.LIB/QTCP.LIB/*, and then remap the temporary bogus URL back to /QSYS.LIB/QTCP.LIB/*. This looks similar to the following:

```
Map   /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/* /DB2WWWBOGUS/*
Fail  /QSYS.LIB/QTCP.LIB/*
Map   /DB2WWWBOGUS/* /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/*
Exec  /QSYS.LIB/QTCP.LIB/*
```

Examples:

- For http://QSYS.LIB/QTCP.LIB/DB2WWW.PGM/... this is mapped to /DB2WWWBOGUS. The Fail does not happen. The next Map fixes the bogus URL. And the final Exec allows the DB2WWW.PGM to execute.
- For http://QSYS.LIB/QTCP.LIB/FTP.PGM/... this does not do the first Map and subsequently fails with the next directive.
- For http://qsys.lib/qtcp.lib/ftp.pgm/... this does not Map, Fail, or Exec due to any of the preceding directives because they are case sensitive. So, it falls through all the preceding directives and eventually fails after all of the directives had been exhausted.

8.4 Hypertext Markup Language (HTML)

The Hypertext Markup Language (HTML) is a simple markup language used to create hypertext documents that are platform independent. HTML documents are SGML (Standard Generalized Markup Language) documents with generic semantics that are appropriate for representing information from a wide range of domains. SGML is an international standard for document markup conforming to ISO 8879.

The latest defined version of HTML is HTML2. But work is being done to provide a new HTML+ or HTML3.

The major enhancements of HTML+ or HTML3 over HTML are:

- Split large documents across multiple servers.
- Support for tables.
- Support for mathematical formulae.

HTML is very similar to a computer programming language without being a programming language. There are commands called tags and syntax rules to be observed when writing in HTML. It looks very much the same as Bookmaster or GML and is very easy to learn and use.

HTML documents can be written using any word processor or text editor. The way they look when seen with a Web browser, however, is quite different from what the writer sees when editing them. It is not a *What You See Is What You Get* (WYSIWYG) approach. There are HTML editors currently available on the market that can be helpful and productive to use when writing your HTML documents.

HTML language provides support for the following features:

- Hypertext links to resources (documents, multimedia, or data files)
- Menu and forms
- Inline graphics
- Text formatting

8.4.1 The HTML Document Structure

The HTML documents are composed of two main parts.

A head Every document should start with a head. The head is the top part of the document. In general, it includes the document title. Why is it, then, important to define the head part in the document?

The different browsers on the market have different ways to display the document's title. The title is also the way by which documents are referenced when saved in the Hotlist or Quicklist of the browser. The title, therefore, must be descriptive but short so it fits into one line of the Quicklist window.

The head of a document cannot contain anchors, any kind of highlighting, or paragraphs. The tags to enclose the head are <HEAD> and </HEAD> (simply meaning start and end of head).

A body The body is the core part of the HTML document. It contains all of the information that is part of the document and controls the way this is presented to browser users.

The body can contain images, lists, menus, entry fields, plain text, or link to other resources. The tags to enclose the body are <BODY> and </BODY> (simply meaning start and end of body).

Figure 139 shows an example of an HTML document.

```
<HTML>
<HEAD>
<TITLE>Ordering an AS/400</TITLE>
</HEAD>
<BODY>
<IMG ALIGN=middle SRC="file:///htmldoc/as400.gif">
<B>PLEASE FILL IN THE FOLLOWING :</B>
<P>
<FORM METHOD="GET" ACTION="/QSYS.LIB/HTML.LIB/ORDAS400.PGM">
Name:
<INPUT TYPE="TEXT" NAME="NAME" SIZE="30" MAXLENGTH="40">
<P>
Address:
<TEXTAREA NAME="ADDRESS" ROWS=2 COLS=30>
</TEXTAREA>
<P>
<B>Which AS/400 would you like to order ?</B>
<BR>
<INPUT TYPE="RADIO" NAME="P1" VALUE="P1">Portable
<INPUT TYPE="RADIO" NAME="P1" VALUE="P2">Server
<INPUT TYPE="RADIO" NAME="P1" VALUE="P3">System
<P>
<B>Do you want the Support Line Service ?</B>
<BR>
<INPUT TYPE="CHECKBOX" NAME="w1" VALUE="w">Yes
<INPUT TYPE="CHECKBOX" NAME="w2" VALUE="f">No
<P>
Thanks for ordering
<INPUT TYPE="SUBMIT" VALUE="Order">
<INPUT TYPE="SUBMIT" VALUE="More Information">
<INPUT TYPE="SUBMIT" VALUE="Cancel">
</FORM>
</BODY>
</HTML>
```

Figure 139. HTML Document

If the HTML document is displayed with a browser, for example, Web browser for OS/2, you see that the title in the head part is displayed in the title bar of Figure 140 on page 186.

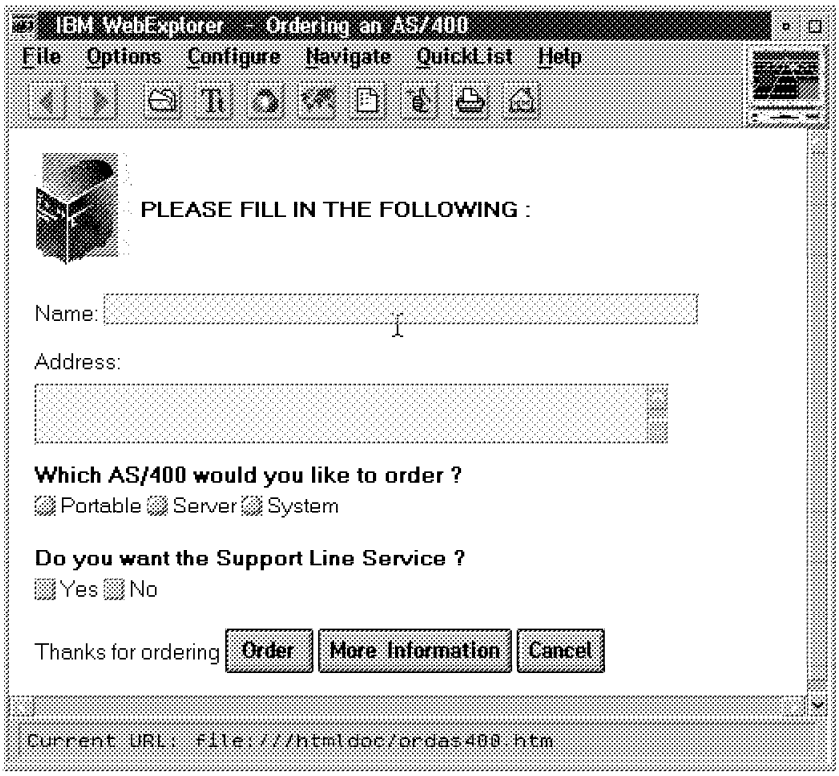


Figure 140. HTML Document Displayed Through an OS/2 Web Browser

8.4.2 HTML Syntax

Let's give you an introduction to the HTML syntax in short terms. The HTML language consists of markup tags to identify the elements of the document. All tags begin with a left angle bracket (<) and end with a right angle bracket (>). Almost every tag is a container. This means that there is always an opening tag and a closing tag the same as in the head and body. Table 9 lists the main HTML elements:

Table 9 (Page 1 of 3). HTML Main Elements

Name	Opening tag	Closing tag	Description
Anchor	< A >	< / A >	HyperLink to a resource.
Address	<ADDRESS>	</ADDRESS>	Format an address.
Bold	< B >	< / B >	Display test in bold.
Base	<BASE>	No closing tag	Record URL of document.
Body	<BODY>	</BODY>	Contain the document's body.
Blockquote	<BLOCKQUOTE>	</BLOCKQUOTE>	Include text in quotes.
Line Break	< B R >	No closing tag	Break current line.
Citation	<CITE>	</CITE>	Specify a citation.
Code	<CODE>	</CODE>	Enclose an example of code.

Table 9 (Page 2 of 3). HTML Main Elements

Name	Opening tag	Closing tag	Description
Definition list description	<DD>	No closing tag	Description of Definition list item
Directory List	<DIR>	</DIR>	Enclose a directory list.
Definition List	<DL>	</DL>	Enclose a list of terms and definitions.
Definition list item	<DT>	No closing tag	Item of definition list
Emphasis	<EM>	</EM>	Emphasize enclosed text.
Form	<FORM>	</FORM>	Define form of enclosed text.
Level 1 heading	<H1>	</H1>	Enclose level 1 heading.
Level 2 heading	<H2>	</H2>	Enclose level 2 heading.
Level 3 heading	<H3>	</H3>	Enclose level 3 heading.
Level 4 heading	<H4>	</H4>	Enclose level 4 heading.
Level 5 heading	<H5>	</H5>	Enclose level 5 heading.
Level 6 heading	<H6>	</H6>	Enclose level 6 heading.
Head	<HEAD>	</HEAD>	Define the head of the document.
Horizontal Rule	<HR>	No closing tag	Insert horizontal line.
HTML	<HTML>	</HTML>	Define HTML document.
Italics	<I>	</I>	Italicize enclosed text.
Image	<IMG>	No closing tag	Embed an image.
Input	<INPUT>	</INPUT>	Display entry field.
Index	<ISINDEX>	No closing tag	Define searchable URL.
Keyboard	<KBD>	</KBD>	Indicate user typed text.
List item	<LI>	No closing tag	Item of Directory list, Menu list, Ordered list, Unordered list
Link	<LINK>	No closing tag	Describe relationship between documents.
Menu	<MENU>	</MENU>	Enclose a menu list.
Ordered List	<OL>	</OL>	Enclose an ordered list.
Option	<OPTION>	No closing tag	Indicate one choice in a Select menu.
Paragraph	<P>	</P> (optional)	Define a paragraph.

Table 9 (Page 3 of 3). HTML Main Elements			
Name	**Opening tag**	**Closing tag**	**Description**
Pre-formatted text	<PRE>	</PRE>	Enclose pre-formatted text.
Sample	<SAMP>	</SAMP>	Indicate sample text.
Select	<SELECT>	</SELECT>	Define a set of selectable options.
Strong Emphasis	<STRONG>	</STRONG>	Strongly emphasize text.
Title	<TITLE>	</TITLE>	Define document's title.
Typetype	<TT>	</TT>	Display enclosed text in monospaced font.
Textarea	<TEXTAREA>	</TEXTAREA>	Enclose a text area.
Underlined	< U >	< / U >	Underline text (unsupported by Mosaic).
Unordered List	< U L >	< / U L >	Enclose an unordered list.
Variable	<VAR>	</VAR>	Indicate a variable.

HTML tags are case insensitive. Every command is interpreted by the browsers independently of the capitalization. The tag <FORM> , for example, can either be written: <Form> or <form> or <fORm> without making any difference.

Note This!

If you want to know more about the Hypertext Makeup Language (HTML), there is a lot of information available on the Internet. There are also a lot of publications available. One of them is the redbook, *Using the Information Super Highway*, GG24-2499.

The most commonly used HTML tags are the headings, list, anchors or links, images, and form tags.

The form tag is used when requesting CGI programs as we shall see in 9.1, "Common Gateway Interface Programs" on page 191, and the image tag is, as it says, used when referring to images or clickable images. See 9.4, "Server Side Image map Support" on page 260 for more information.

8.4.3 Logging the Access of the Web server

Internet Connection/400 has the ability to capture client information whenever a client utilizes your World Wide Web server or HTTP server. For example, information is stored in a database for ease of access by the marketing division for future business contacts. This allows you to use existing query and reporting tools or the direct database serving feature, DB2WWW, of Internet Connection/400 to provide up-to-the-minute reports detailing client usage.

The information stored in the database is:

URL address of the page or pages accessed
This gives you invaluable information about the clients' preferences and interests by storing the address of each page accessed.

IP address of clients

This information identifies the location of the clients. You can find out where the requests for information originated. This data helps identify clients whether they are suppliers, competitors, future customers, or present customers.

Date and time of access

The information provides the exact date and time of each access of the server, which allows you to allocate resources to handle a client′s demand for information.

Logging of the World Wide Web server access notifies you about what information is being accessed, when that information is requested, and by whom. In fact, it is extremely helpful if you want to use the Internet for marketing purposes.

If the ultimate goal of logging remote IP addresses is to build up some kind of E-mail mailing list, you find that this does not work too well. For one thing, the hits to your web pages could be coming from a user sitting behind a firewall. In this case, the remote IP address that the AS/400 HTTP server sees is the firewall′s, not the users. Also, many people connect to the Internet through temporary IP addresses given to them by their ISP. In both of these cases, it is difficult to contact the original person that was so interested in your products.

That is why so many web pages have a CGI-bin form that asks the person′s E-mail address and some other form of identification. When (and if) the customer completes the form and sends it in to your AS/400 system, the CGI-bin application that is called can add a new record to a mailing database file, for example.

As mentioned earlier, we have included into library ITSOIC400 the object and source code for program PRTWWWLOG, which is a quite simple ILE RPG program that produces a print of the AS/400 system′s WWW access log. It requires file WWWLOG to be overridden to the appropriate log file and member with the OVRDBF command.

8.5 I/NET′s Web Server/400

I/NET′s Web Server/400, Commerce Server/400 and Webulator/400 are three other web server products available, at a cost, for the AS/400 system.

Figure 141. I/NET's Home Page at http://www.inetmi.com

I/NET's HTTP servers are fully HTTP/1.0 compliant and as this book is being written, offer the following features:

- Integrated File System support
- Scripts (including REXX, CL, ILE-C, RPG, and COBOL)
- Database, shared folders, and spooled files
- Ultimedia System Facilities support
- Logging
- National language support

In addition, I/NET's Commerce Server/400 includes encryption technologies based upon the SSL standard for the purpose of conducting secured commerce across the Internet.

The Webulator/400 is a 5250 to HTML gateway product that is similar to IBM's 5250 Work Station Gateway.

For the latest details of their product, take a look at URL: `http://www.inetmi.com`

Chapter 9. Application Development Interfaces for World Wide Web

This chapter is broken up into four wide pieces. It starts out with the Common Gateway Interface (CGI) which is one of the standard ways that HTTP servers can pass parameters to and from the web clients. CGI programming gives you great flexibility, but to program something more than just a simple single-screen application can be very difficult and time consuming.

The next section shows us how to use the DB2WWW macro interface (now being called net.data) to access SQL data and produce HTML reports. This greatly eases the burden on the programmer, but is not as flexible as CGI.

The third section shows us the HTML to 5250 Gateway support which is by far the easiest way to create web applications on the AS/400. Because the conversion to and from HTML from 5250 is handled automatically by the AS/400 most application do not have to change to take advantage of this.

This chapter closes with a look at how to code server side image maps and with a list of additional publications and resource available to you on the web.

9.1 Common Gateway Interface Programs

Important note

If you would like to follow along with the examples in this and other chapters you first will want to install the ITSO Company demonstration and other web based applications from the CD-ROM that came with this redbook. Please see Appendix A, “Installing the ITSO Company Demo” on page 285 for instructions on how to get your AS/400 up and running right away.

As an owner of an HTTP server on the Internet, you may sometimes see an advantage in allowing the readers of your documents (HTML documents) to return information back to your server, or let the readers retrieve certain kinds of information from your server through the execution of special programs.

The mechanism for doing these or other similar functions is called the Common Gateway Interface (CGI).

A programmer creates a gateway program and places it in the /CGI-BIN directory of the server. A user of a Web page selects the gateway program URL as part of using a Web page link, FORM, or ISINDEX query box. In response to the user request, the gateway program executes and returns the results to the client.

The (CGI) is a standard for interfacing external applications with information servers, such as HTTP servers or Web servers. The current version is CGI 1.1.

The basic idea of the CGI is illustrated in Figure 142 on page 192.

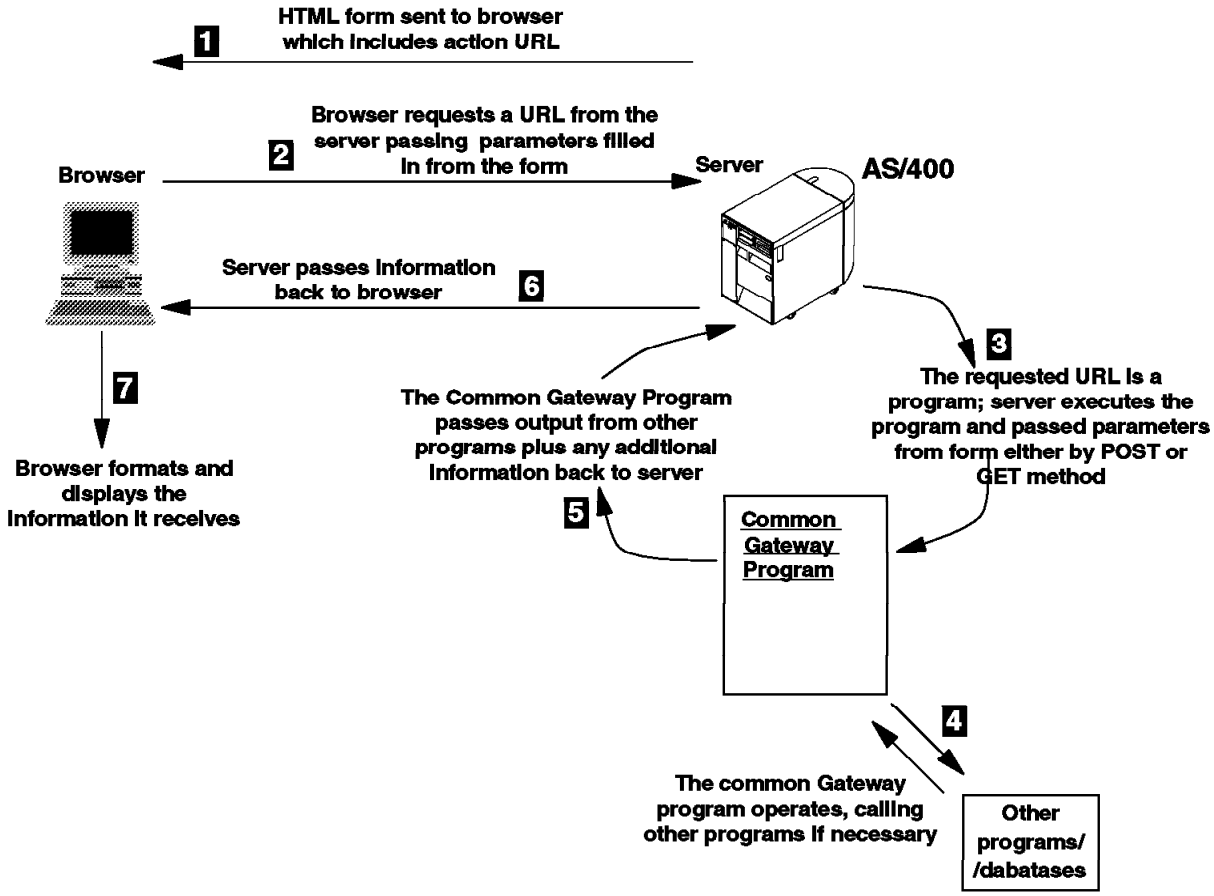


Figure 142. The Basic Idea of How the CGI is Working

1. An HTML form is sent from the server to the web client. Along with the form is an action URL that points back to the CGI-bin program located someplace on the network.
2. The browser is sending a request for a URL (Uniform Resource Locator) resource from the server. This URL contains the name and path of the CGI-bin application along with all of the form's parameters.
3. The requested URL is a program - a Common Gateway Interface program. The parameters are passed to the application program by one of two ways depending on the form's method of either post or get.
4. The server is executing the CGI-program. While running, the CGI-program is calling other programs or databases.
5. The CGI program passes the output from the other programs or databases plus any additional information back to the server. This is done by writing to the standard output of the application (stdout).
6. The server passes the information back to the browser. This information is probably more HTML or a redirection indicator.
7. It is the browser's job to format the information it receives and display it.

URL

The URL specifies the address or location of a resource, such as CGI programs, on the Internet. The URL looks similar to this:

protocol://host_name:port/pathname (?parameters)

To do all of this, a set of standards has been set on how to implement CGI programs on your server so every possible web browser that exists can access your server and execute the CGI program.

9.1.1 Programming

For accessing a CGI program, you have to use the FORM tag in your HTML document. Figure 143 shows an example on how to write an HTML document with the FORM tag.

```
<html>
<head>
<title>Ordering an AS/400 Method=GET</title>
</head>
<body bgcolor="#F8F8FF">
<img align=middle src="as400.gif">
<b>    Ordering an IBM System AS/400 </b>
<p>
<b>Please fill in the following :</b>
<form method="GET" action="/BonusCGI/ORDAS400G.PGM">
<INPUT type="hidden" name="MBR" value="ORDAS400E " size=10>
Name :
<input type=text name="NAME" size="30" maxlength="40" clear="all">
<p>
Address :
<textarea name="ADDRESS" cols=30 rows=2 VALUE=" ">
</textarea>
<p>
<b>Which AS/400 would you like to order ?</b>
<br>
<input name="TYPE" value="P1" type=radio checked>Portable
<input name="TYPE" value="P2" type=radio>Server
<input name="TYPE" value="P3" type=radio>System
<p>
<b>Do you want the Support Line Service ?</b>
<br>
<input name="SERV" value="T" type=radio checked>Yes
<input name="SERV" value="F" type=radio>No
<p>
Please order :
<p>
<input type=submit VALUE="Order">
<input type=reset VALUE="Clear Form">
</form>
</P>
<A HREF="/web/DEMO2E.MBR">Back</A> Demo-Menue
```

Figure 143. The HTML Source Using the FORM Tag

When the Web browser displays the HTML document, it is similar to the following figure.

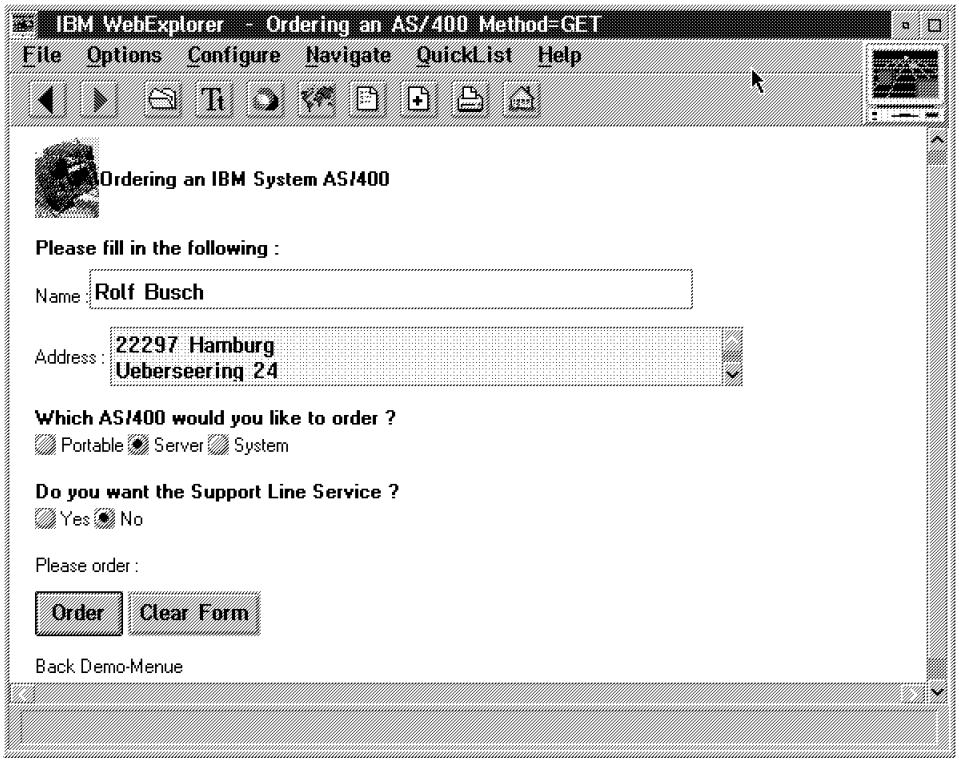


Figure 144. The HTML Document Using the FORM Tag Displayed by the Browser

A Form basically consist of three elements:

1. The Form method:

   ```
   <FORM METHOD="GET" ACTION="/BonusCGI/ORDAS400G.PGM">
   ```

   or

   ```
   <FORM METHOD="POST" ACTION="/BonusCGI/ORDAS400P.PGM">
   ```

2. The Input tags:

   `text, textarea, checkbox, radiobutton, options...`

3. Submit button

All three elements are used in Figure 143 on page 193. Note that the *action* keyword contains the URL address to send the input data to when the submit button is pressed. In this case, *action="/BonusCGI/ORDAS400x.PGM"*.

A match with an EXEC directive in our HTTP configuration enables the HTTP server to execute a CGI program on behalf of the client.

In our case, action="/BonusCGI/ORDAS400x.PGM" is mapped with:

```
EXEC /BonusCGI/*  /QSYS.LIB/ITSCOIC400.LIB/*
```

Note This!

Please note that only programs in the QSYS.LIB library may be the target for the input data.

This is because only programs in the QSYS.LIB File System may be executed on the AS/400 system.

The following figure shows the HTML output after the execution of the CGI-BIN program.

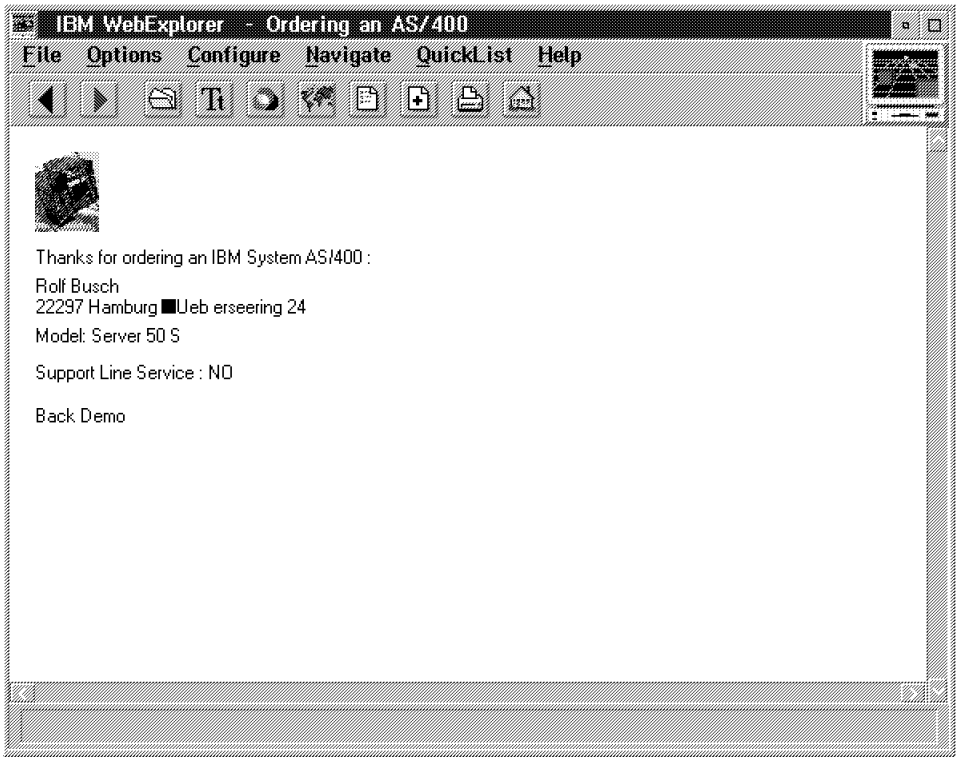


Figure 145. The HTML Output Information Back on the Browser

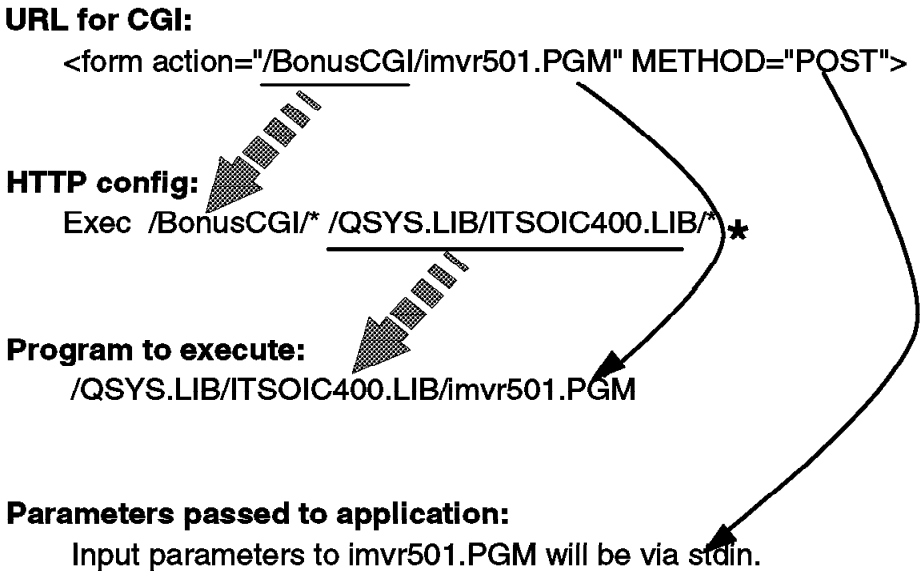


Figure 146. Mapping of CGI URL to HTTP Configuration

There are two *methods* that can be used to access your forms. Depending on which method you use, you receive the encoded results of the form in a different way. The two methods are:

Method	Description
GET	If your form in the HTML document has METHOD="GET" in its FORM tag, your CGI program receives the encoded form input in the environment variable QUERY_STRING.
POST	If your form in the HTML document has METHOD="POST" in its FORM tag, your CGI program receives the encoded form input on stdin. The server does *not* send you an End of File (EOF) on the end of the data. Instead, you must use the environment variable CONTENT_LENGTH to determine how much data you should read from stdin.

Note This!

POST is often the preferred method of operating with CGI-BIN programming.

The reason why POST is preferred is that GET uses the environment variable QUERY_STRING.

Some server platforms are limited in the length of environment variables.

A Form with a lot of data could cause a loss of data.

In the POST method, there are no limits (all date is passed through STANDARD INPUT, a way of transferring information in a "stream" without limits).

As you can see, the form method determines the way of passing the encoded input data from the browser to the CGI program through the server. The server and the CGI program communicates in four major ways, all supported by the AS/400 system:

Environment variables	Used for the GET method.
The command line	Used in the case of an ISINDEX query.
Standard input	Used for the POST method.
Standard output	Used to produce output data.

The AS/400 HTTP server provides the means whereby called CGI programs can access environment variables and standard input stream (stdin). The server also provides the means for a CGI program to return data in the standard output stream.

Request for programs in an HTML document can be made using either the GET or the POST method. See Figure 147 on page 197 and Figure 148 on page 199.

Note This!

The AS/400 system runs CGI programs under the QTMHHTP1 user profile. The QTMHHTP1 user profile **must** have authority to access all the objects accessed by the CGI programs.

9.1.2 GET

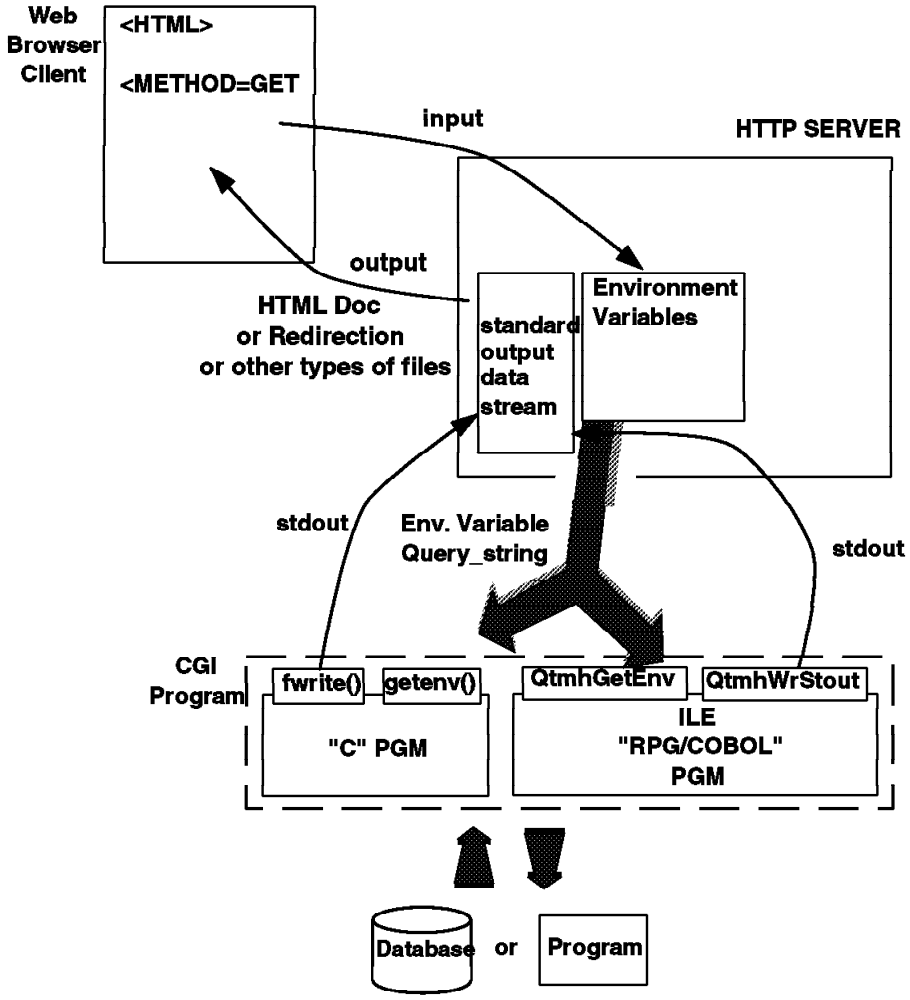


Figure 147. AS/400 CGI Using the GET Method

As you can see in Figure 147, the flow is the same as in the basic chart for CGI Figure 142 on page 192. The HTML document displayed by the browser uses `<FORM METHOD="GET".`

When the `GET` method is used, the input parameters are passed to the called CGI program in one of two ways:

1. In command line parameters (program call parameters):

   Parameters are passed to the CGI program using command line arguments (program call parameters) only when the data in the request URL beyond the first question mark (?) contains *no* equal signs (=). This is called an `ISINDEX`. In the URL for an `ISINDEX` request, the CGI parameters are positional (being separated by plus (+) signs). The server parses the parameters and puts each parameter, according to its position in the string, into corresponding parameters of the CGI program call.

   An example of a URL is:

http://sysnm003.sysnmaa.ibm.com/CGILIB/INDEX.PGM?value1+value2+value3

2. In the environment variable QUERY_STRING:

   If the parameter string of the URL beyond the first question mark (?) contains an equal sign (=), the entire parameter string is put into the QUERY_STRING environment variable. This is the most normal case.

   An example for the URL is:

   http://sysnm003.sysnmaa.ibm.com/CGILIB/FORM.PGM?kwd1=value1&kwd2=value2&kwd3=value3

Note

The assumption is made that CGILIB has been mapped to a library in QSYS.LIB containing CGI programs with the MAP and Exec directives for the AS/400 HTTP server.

The mapping directives allow the Web Administrator on the AS/400 system to define a virtual hierarchy of web resources that is presented by the AS/400 HTTP server. This virtual hierarchy is independent of the physical file system (or systems) on which the resources are actually stored. This is very useful since:

1. It allows resources to be physically relocated without changing the apparent hierarchy and its implied associations.
2. The details of underlying physical file systems can be hidden from client applications that are accessing the AS/400 HTTP server.

The second reason is especially important for the AS/400 HTTP server because of the differences between some of the AS/400 file systems and the tree structured hierarchical UNIX file systems upon which the HTTP protocol and the URL scheme are based.

The mapping directives, MAP and Exec, are used to specify a set of rules that define a process for translating and processing the path (and file) information contained in a URL that is received in a client request URL.

Each of the mapping directives may be specified multiple times and in any combination, and may be placed anywhere in the AS/400 HTTP server configuration. When the AS/400 HTTP server configuration is read during initialization, the mapping directives are stored in order of appearance in a mapping rule table.

The incoming URL is compared against each of the specified mapping rules in turn. If a directive applies, the specified translation is carried out. Depending on the rule that was matched, rule processing is either suspended or continues with the next rule using the newly translated URL.

The mapping and the exec definitions are created through the WRKHTTPCFG command.

9.1.3 POST

When the POST method is being used, the QUERY_STRING environment variable contains a null string. The null string makes the CGI program look for its input in the standard input stream. See Figure 148. The CGI program can determine the method, GET or POST, from the REQUEST_METHOD environment variable.

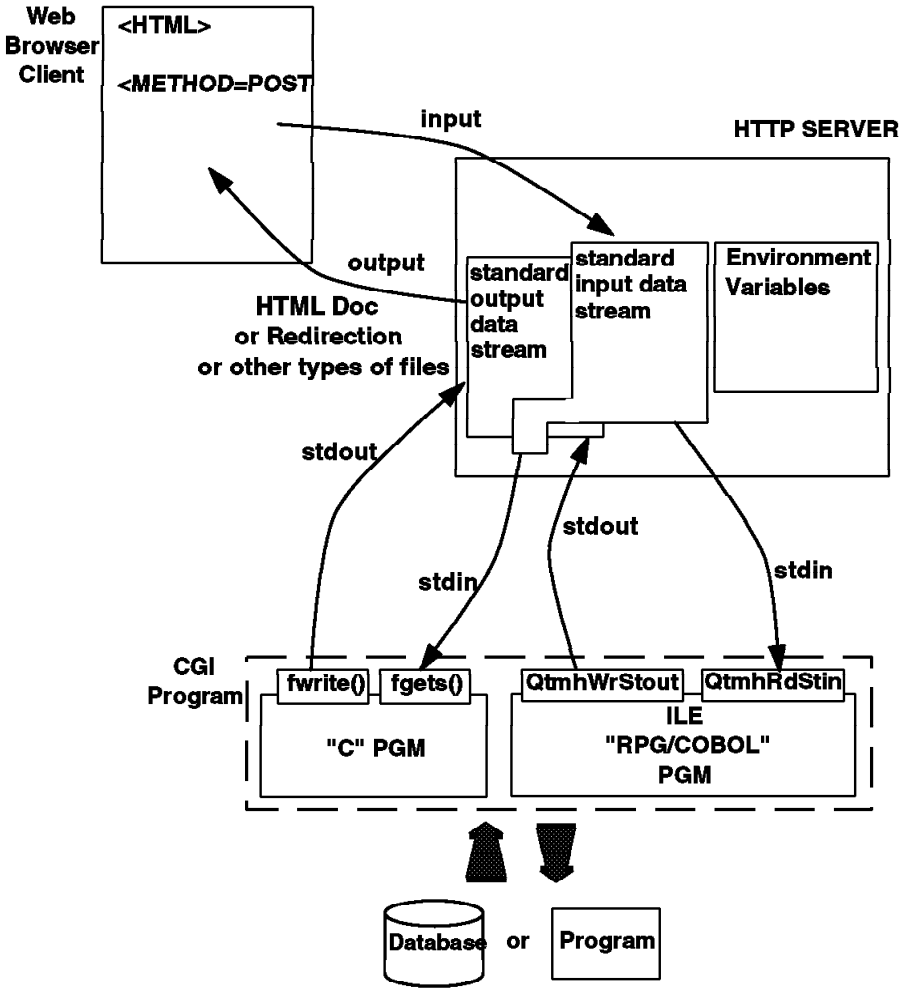


Figure 148. AS/400 CGI Using the POST Method

When the HTML document uses the <FORM METHOD="POST", the form is submitted with the standard input data stream, stdin. The CGI program inspects the request method to determine if post data is available by reading the REQUEST_METHOD environment variable. The CGI program can obtain other environment variables set by the HTTP server regardless of the request method.

In order to support environment variables, the AS/400 HTTP server creates an environment variable user index that contains the name and value for each environment variable set by the server. The index is named QTMHENVI in the QTEMP library.

The reason why it is created in the QTEMP library is that it has to be unique for each child job that defines it. So each child job creates the index that is accessible to the CGI programs that the server child job calls.

Each time a CGI program is called, a new index is created and new environment variables are set (preventing one CGI program from using the environment variables intended for another CGI program). Creation of the index is not dependent on the use of either method, GET or POST. The index is always created.

The QTMHENVI index contains the following environmental variables:

Table 10. CGI Environment Variables in the QTMHENVI Index File

Environment Variable	**Contain**
QUERY_STRING	Information that follows the first question mark (?) in the URL referencing the CGI program.
SERVER_SOFTWARE	The name and version of the information server software answering the request (and running the gateway).
SERVER_NAME	Host name, DNS alias, or IP address of the server.
GATEWAY_INTERFACE	The version of the CGI specification to which the server complies.
SERVER_PROTOCOL	The name and revision of the information protocol this request came in with.
SERVER_PORT	The port number to which the request was sent.
REQUEST_METHOD	The method with which the request was made. For HTTP, this is ″GET″ or ″POST″.
PATH_INFO	The extra path information following the path information required to identify the CGI program name.
PATH_TRANSLATED	The server provides a translated version of PATH_INFO, which takes the path and does any virtual-to-physical mapping to it.
SCRIPT_NAME	A virtual path to the program being executed.
REMOTE_HOST	The host name making the request.
REMOTE_ADDR	The IP address of the host name making the request.
REMOTE_USER	This is the user name making the request.
CONTENT_TYPE	For queries that have attached information, such as HTTP POST, this is the content type of the data.
CONTENT_LENGTH	The length of data in the attached HTTP POST from the client.
IBM_CCSID_VALUE	The CCSID under which the current server job is running.

When the CGI program has finished executing, it is expected to return an HTML document or a redirection to an HTML document.

The CGI program passes output from other programs or databases plus any additional information back to the server. To do that, the CGI program uses the standard output data stream, stdout.

There is support for CGI programs written in:

- C language
- ILE RPG
- ILE COBOL

Necessary APIs are available to support the environment variables, the standard input stream, and the standard output stream.

The C language supports the environment variables through the getenv() API. The standard input stream and the standard output stream are supported through many APIs. Examples are fgets() and fwrite().

For the C programs, the APIs are found in the QTMHCGI *SVCPGM. A C language header file member named QTMHCGI is provided in H file in library QSYSINC.

The APIs to support ILE RPG and ILE COBOL are provided in source files. The source files are:

- QCBLLESRC for ILE COBOL
- QRPGLESRC for ILE RPG

four APIs are provided:

Table 11. The APIs Provided for ILE RPG, ILE COBOL, and C Programs

Type of API	ILE RPG	ILE COBOL	C Language
Get Environment Variable	QtmhGetEnv	QtmhGetEnv	getenv()
Read from Stdin	QtmhRdStin	QtmhRdStin	Many, for example fgets()
Write to Stdout	QtmhWrStout	QtmhWrStout	Many, for example fwrite()
Parse CGI input	QtmhCvtDb	QtmhCvtDb	#pragma mapinc

Note this for RPG!

For the **CRTPGM**, you **must** always specify the BNDSRVPGM **QTCP/QTMHCGI** to ensure the module import for QtmhGetEnv, QtmhRdStin, QtmhWrStout, and QtmhCvtDb are satisfied. If you forget to include the SRVPGM you will receive something like this message:

```
Definition not found for symbol 'QtmhRdStin'.
```

Case sensitive alert!

The service program APIs of QtmhGetEnv, QtmhRdStin, QtmhWrStout, and QtmhCvtDb are case sensitive. If you misspell one of them or if the case is wrong for just one of the letters this is the message you will get after your CRTPGM (the binding step) fails:

```
Message ID . . . . . :   CPD5D1D       Severity . . . . . :   20
Message type . . . . . :   Diagnostic
Date sent  . . . . . :   10/30/96      Time sent  . . . . . :   14:33

Message . . . . :   *SRVPGM object QZDMMDTA in library QSOC not found.
Cause . . . . . :   *SRVPGM object QZDMMDTA in library QSOC was specified
  binding directory QUSAPIBD in library *LIBL, but was not found for bind
Recovery  . . . :   Contact your application provider or service
  representative.
```

This message is seemingly not related at all to the source of the problem.

The APIs are provided to simplify parsing form data from either stdin or from the environment variable QUERY_STRING. The CGI program is expected to provide the name of the DDS file specification that identifies the anticipated form variables and their attributes.

The server times the operation CGI programs and terminates child server jobs exceeding the time limit. There are three timeout keywords for input timeout, output timeout, and script timeout.

Also, the server protects itself from CGI program exceptions by providing exception handling that reduces any escape messages resulting from CGI program failures to diagnostic messages.

9.1.4 Decoding the Parameters from the Remote Web Client

You now know that depending on the HTML Form′s method, which could be either POST or GET, the input string to your application will made available to your CGI application running on the AS/400 in either standard in or the QUERY_STRING environment variable. If your application is in ILE C/400, then you have native I/O statements to code to retrieve the parameter string. If your application is in ILE COBOL/400 or ILE RPG/400, IBM provides a service program to do the job of reading the parameter string and placing it in a local program variable.

What we have not covered is what this parameter string looks like and what exactly you must do to parse it into something that you can use in your application. How to parse the string is dependant upon if you are writing your CGI application in C/400 (see 9.1.4.2, “CGI Parameter Parsing with ILE C/400” on page 204), or in either RPG/400 or COBOL/400 (see 9.1.4.3, “CGI Parameter Parsing with ILE COBOL/400 or RPG/400” on page 204).

But first, let′s spend a little bit of time to understand the basic syntax of the parameter string that we will receive from a web client.

9.1.4.1 CGI Parameter String Syntax

When you write the HTML form, each of your input items has a NAME tag associated with it. As an example, take a look at the FORM tag that we find in library ITSOIC400, file HTML and member FORMGETE (see Figure 149). The first input field is a hidden field which simply means that the end user will not see this field nor will he be given the opportunity to modify the value associated with it. The second named field is ″querydata″ which will be presented to the end user with the possibility to update. The input that is entered by the end user for each input field is called the value.

```
<FORM METHOD=″GET″ ACTION
  =″/BonusCGI/CGIENVGET.PGM″>
<INPUT type=″hidden″ name=″MBR″ value=″CGIENVGETE″ size=10>
<INPUT TYPE=text NAME=″querydata″ MAXLENGTH=″20″ VALUE=″ ″>
<BR>
<BR>
<INPUT TYPE=″SUBMIT″ VALUE=″Send Request″>
<INPUT TYPE=″RESET″ VALUE=″Clear Your Input″>
</FORM>
```

Figure 149. Two Named Input Fields Defined in the HTML Form

It is this stream of name=value pairs that flow back to your CGI application running on the AS/400 when the user selects the submit button with the mouse. These are the name=value pairs that you will be reading either from standard in or the QUERY_STRING environment variable.

One more thing must be mentioned. A well behaved web client will modify the format of the parameter string. Each name=value pair will separated by the ampersand (&) character. Also, each name=value pair is URL encoded, which means spaces are changed into pluses (+) and some characters are encoded into hexadecimal.

Again, using the same form as seen in Figure 149 let us assume that the end user enters a value for querydata of:

```
AaBbCc, First three
```

The content of QUERY_STRING, as echoed back to the client by CGI program CGIENVGET (also found in the ITSOIC400 library), can be seen as:

```
MBR=CGIENVGETE&querydata=AaBbCc%2C+First+three
```

Let's parse this string ourselves:

```
MBR=CGIENVGETE&querydata=AaBbCc%2C+First+three
|              |              |  |     |
|              |              |  Note : Spaces become plus (+) signs
|              |              Special characters, like the comma (,)
|              |              are represented as three ASCII character
|              |              escape sequences representing a hexadecimal
|              |              entry into an ASCII code page.
|             Second name=value pair.  The querydata name is case
|             sensitive.  The value follows the equal (=) sign.
|
First name=value pair.  The name is MBR and is case sensitive.  The
value follows the equal (=) sign.
```

For more information about the format of the name=value pairs we suggest that you look on the web. A good place to start is http://hoohoo.ncsa.uiuc.edu/cgi/.

To learn how to parse the name=value pairs coming from the web client, keep reading.

9.1.4.2 CGI Parameter Parsing with ILE C/400

If you are using ILE C/400 as your CGI application on the AS/400, you have one advantage. And that advantage is that many other platforms, namely UNIX based, also use C as a tool for writing CGI. And, if you have not figured it out yet, you will soon realize that the CGI interface was born and develop on UNIX styled systems, first. So, for the job of parsing the name=value pairs that come from the web client we can go to the web to pull down public domain C code examples. This is what we did.

The source of the parsing routines can be found in library ITSOIC400, file UTILITY, members UTIL (the C source code) and UTIL_H (the header include files). The UTIL code has been compiled and made into a service program that you may bind to your own ILE C/400 CGI applications - or to the one that we provide as a sample in the ITSOIC400 library. For this example, please see 9.1.6.3, “Source-Code C Program PARSECGIP” on page 214.

We did need to modify the CGI parsing algorithm to work on the AS/400. The problem porting these routines came when the three character escape sequences (like %2C that represents the comma (,) character) where being translated into a single character. The code that we found on the web assumed that the native character set on the system it was running was ASCII. The AS/400 is EBCDIC, so we needed to modify the routine once we understood the problem.

This, as we will see in 9.1.4.3, “CGI Parameter Parsing with ILE COBOL/400 or RPG/400,” is a problem in more than just this one place.

9.1.4.3 CGI Parameter Parsing with ILE COBOL/400 or RPG/400

IBM has provided a service program routine by the name QtmhCvtDb to automatically parse the name=value string and place the results in an externally defined physical file. Externally defined by DDS, of course. This makes the job of parsing this rather complex string very easy for ILE COBOL/400 and ILE RPG/400 programmers as it is trivial to define and then read in the fields associated with the DDS file in RPG, as an example. This is the good news.

The bad news is that the QtmhCvtDb as shipped with V3R2 of the TCP/IP Connectivity Utilities/400 has a bug in it (at the time of the writing of this redbook) that causes it to make a mistake in the translation of the special escape sequences (like %2C) into the EBCDIC equivalent. The root cause of the mistake lies in the same problem described in 9.1.4.2, “CGI Parameter Parsing with ILE C/400.”

Here are some circumventions to the problem that can be used until IBM fixes QtmhCvtDb:

1. Use the service program UTIL as mentioned in 9.1.4.2, “CGI Parameter Parsing with ILE C/400” to parse the string of name=value pairs. The source of this code is provided in the library ITSOIC400, file UTILITY, member UTIL. This code would be useful if you need to make some modification to the code for your environment. You may find, for example, the translation

table that we use to convert ASCII to EBCDIC may not work for your environment.

2. You can always write the parsing code yourself in any language you want.
3. PID (IBM's Partners In Development) have made available a code snippet (small section of source code) that you can place right before the call to QtmhCvtDb. This code snippet modifies the string *before* the call by changing the hexadecimal characters of the escape sequence that point to a position in an ASCII code page into the EBCDIC representative. This causes QtmhCvtDb to proper convert the escape sequences into a single EBCDIC character.

   When IBM fixes the problem with QtmhCvtDb, you simply remove the code snippet.

   For more information (and the source code) go to URL `http://www.softmall.ibm.com/as400/news2.html`.

9.1.5 Examples for Environment Variables

On the following pages are some inquiry examples for environment variables.

Environment Variables with GET

Figure 151 on page 206 shows the inquiry results with the `Method=GET`.

The environment variable `QUERY_STRING` contains the name/value pairs!

Environment Variables with POST

Figure 153 on page 207 shows the inquiry results with the `Method=POST`.

The environment variable `QUERY_STRING` is **empty**.

`POST` always reads from **STDIN**!

The following figure shows the Form for program `CGIENVGET`: Environment Variables with `METHOD=GET`.

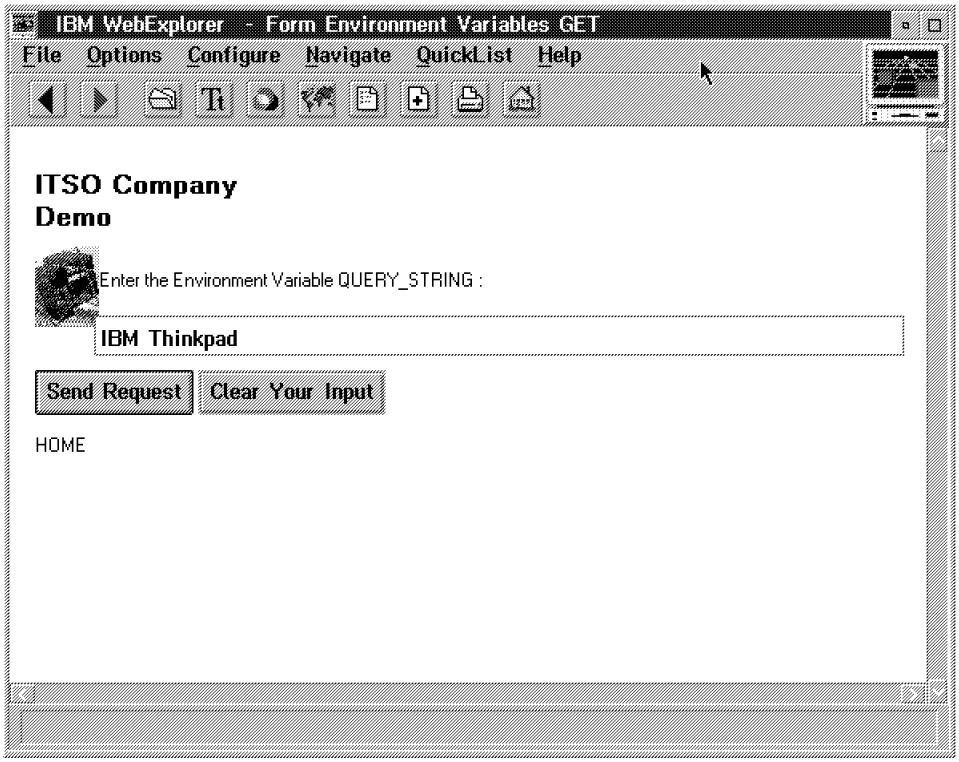


Figure 150. The HTML Document Using the FORM Tag with Method=GET

The following figure shows the HTML output from program CGIENVGET.

Environment Variables with GET

QUERY_STRING :	MBR=CGIEN VGETE&querydata=IBM+Thinkpad+
SERVER_SOFTWARE :	IBM-AS/400-HTTP-Server/1.0
SERVER_NAME :	internut.r chland.ibm.com
GATEWAY_INTERFACE :	CERN-PreParsed
SERVER_PROTOCOL :	HTTP/1 .0
SERVER_PORT :	80
REQUEST_METHOD :	GET
PATH_INFO :	N/A
PATH_TRANSLATED :	N/A
SCRIPT_NAME :	/BonusCGI/CGIENVGET.PGM
REMOTE_HOST :	w3proxy-b. rchland.ibm.com
REMOTE_ADDR :	9.5.100.112
REMOTE_USER :	N/A
CONTENT_TYPE :	application/x-www-form-urlencoded
CONTENT_LENGTH :	45
IBM_CCSID_VALUE :	N/A

Back Demo

Figure 151. Document Returned from the Program CGIENVGET

The following figure shows the Form for program CGIENVPOST: Environment Variables with METHOD=POST.

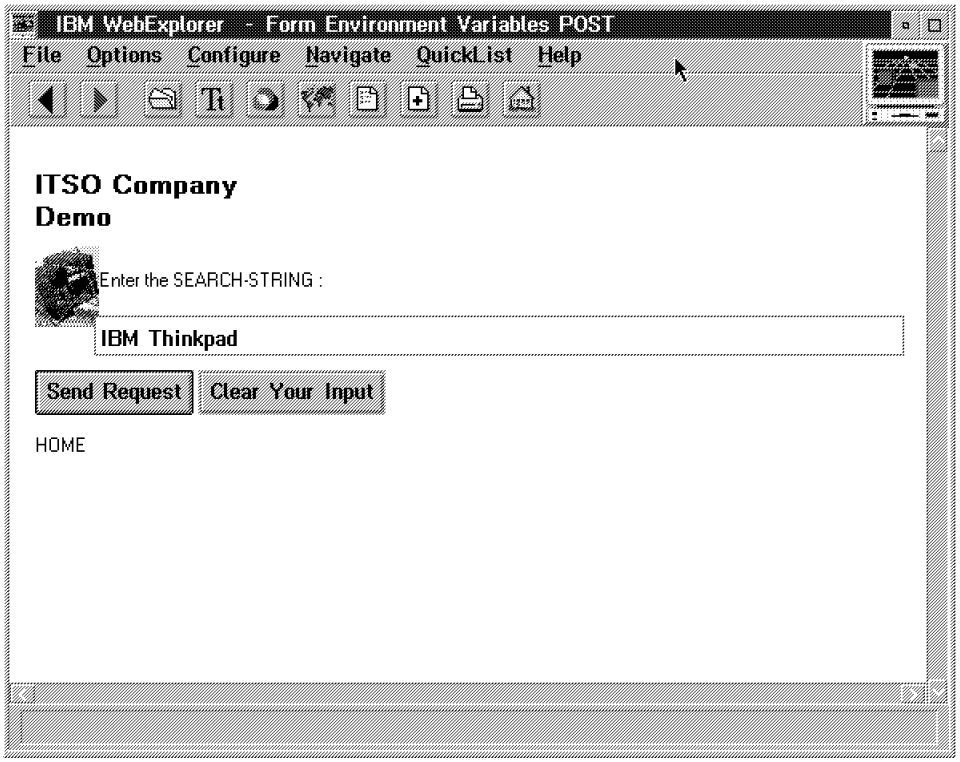


Figure 152. The HTML Document Using the FORM Tag with Method=POST

The following figure shows the HTML output from program CGIENVPOST.

Environment Variables with POST

QUERY_STRING :	N/A
SERVER_SOFTWARE :	IBM-AS/400-HTTP-Server/1.0
SERVER_NAME :	internut.r chland.ibm.com
GATEWAY_INTERFACE :	CERN-PreParsed
SERVER_PROTOCOL :	HTTP/1 .0
SERVER_PORT :	80
REQUEST_METHOD :	POST
PATH_INFO :	N/A
PATH_TRANSLATED :	N/A
SCRIPT_NAME :	/BonusCGI/CGIENVPOST.PGM
REMOTE_HOST :	w3proxy-b. rchland.ibm.com
REMOTE_ADDR :	9.5.100.112
REMOTE_USER :	N/A
CONTENT_TYPE :	application/x-www-form-urlencoded
CONTENT_LENGTH :	38
IBM_CCSID_VALUE :	N/A

Back Demo

Figure 153. Document Returned from the Program CGIENVPOST

9.1.6 ITSO Company Demonstrations

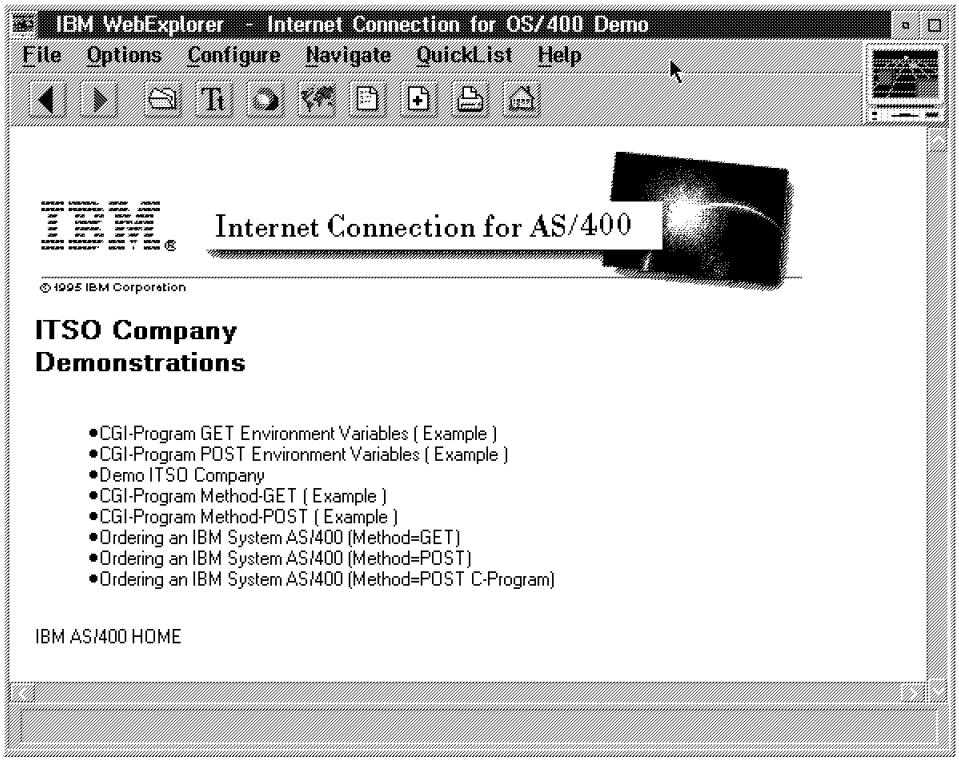


Figure 154. Demonstrations Welcome to the ITSO Company

All programs, files, HTMLs and source files are located in the library ITSOIC400. The images (GIF) are in the directory /ITSOIC.400.

Table 12. Overview Demo - Programs HTMLFILE=INPUT

Program	Function	Files	HTML-Mbr	HTMLO-Mbr	Language
CGIENVGET	Environment Variables GET	ENVFILE ENVGETDS HTMLO	FORMGETE	CGIENVGETE	ILE RPG
CGIENVPOST	Environment Variables POST	ENVFILE ENVPOSTDS HTMLO	FORMPOSE	CGIENVPOSE	ILE RPG
CGIGET	Search Online Catalog	INVFILE QUERYDS	L04P040		ILE RPG
CGIPOST	Feedback Talk To Us	TALKTOUS TALKDS	L04P060		ILE RPG
ORDAS400G	Ordering an AS/400 GET	ORDAS00F ORDASDS HTMLO	AS4FORMGE	ORDAS400E	ILE RPG
ORDAS400P	Ordering an AS/400 POST	ORDAS00F ORDASDS HTMLO	AS4FORMPE	ORDAS400E	ILE RPG
PARSECGIP	Ordering an AS/400 POST		RESIFORM		ILE C

9.1.6.1 Source-Code RPG Program ORDAS400G

ILE RPG program example for the Method=GET: The information from the URL (value/name pairs) is assigned to the environment variable QUERY_STRING.

This gateway program picks out the name/value pairs of the variables from this QUERY_STRING through the API for ILE RPG:

The following example shows the Get Environment Variable QtmhGetEnv (subroutine GETENV).

```
     ****************************************************************************
     *  Simple ILE RPG program ORDAS400G to test CGI Method=GET
     *
     *
     * 1. Compile this source member as module ORDAS400G ( PDM Option=15 )
     *
     * 2. Create program ORDAS400G from module ORDAS400G  ( PDM Option=26 )
     *    with PROMPT(PF4) and BNDSRVPGM(QTCP/QTMHCGI)
     *
     * Define your files here
     ****************************************************************************
     * Order Database file
     ****************************************************************************
    FORDASOOF  O  A E             DISK
     ****************************************************************************
     * HTMLFILE Input file   ( read FORM input )
     ****************************************************************************
    FHTMLFILE  IF   E             DISK
     ****************************************************************************
     * HTMLO   Output file   ( prepared HTML Output in SRC-PF HTMLO  )
     *                       ( MBR = hidden field in HTML Input Form )
     ****************************************************************************
    FHTMLO     IF   E             DISK    USROPN  RENAME(HTMLO:HTMLOUT)
     ****************************************************************************
A
|    *Variables for the CGI interface APIs
|    *These are for the APIEnVar
|   DENBuff           S           2048A   INZ
|   DENBuffLn         S              9B 0 INZ(2048)
|   DENActLn          S              9B 0 INZ
|   DENVarName        S             64A   INZ
|   DENVarLn          S              9b 0 INZ
A
     *These are for the APICvtDB  Datastructure INPUT fields
     ****************************************************************************
    DDBFileName       S             20A   INZ('ORDASDS   *LIBL     ')
     ****************************************************************************
    DDBBuff           S           2048A   INZ
    DDBBuffLn         S              9B 0 INZ(2048)
    DDBDSLn           S              9B 0 INZ
    DDBActLn          S              9B 0 INZ
    DDBRespCd         S              9B 0 INZ
     *These are used for APIStdOut
    DOutBuff          S           2048A   INZ
    DOutBuffLn        S              9B 0 INZ(2048)
     ****************************************************************************
     *Externally described data structure.  Used for Parsing
     *Need a different one in each CGI-BIN you write
    DORDASDS        E DS
```

Figure 155 (Part 1 of 5). ILE RPG Program ORDAS400G Method=GET

```
 ****************************************************************************
 * Data structure for error reporting.  Copied from QSYSINC/QRPGLESRC(QUSEC
DQUSEC            DS
D*                                              Qus EC
D QUSBPRV                1      4B 0 INZ(16)
D*                                              Bytes Provided
D QUSBAVL                5      8B 0
D*                                              Bytes Available
D QUSEI                  9     15
D*                                              Exception Id
D QUSERVED              16     16
D*                                              Reserved
D*QUSED01               17    116
D*
D*                                    Varying length
 ****************************************************************************
 *Constants for names of CGI APIs
DAPIStdIn         C                   'QtmhRdStin'
DAPIStdOut        C                   'QtmhWrStout'
DAPICvtDB         C                   'QtmhCvtDb'
DAPIEnVar         C                   'QtmhGetEnv'
 *Compile-time array for OVRDBF
DOVRDBF           S             80    DIM(2) PERRCD(1) CTDATA
DOVRARR           S              1    DIM(80)
D****************************************************************************
D* Define NewLine
DNewLine          C                   x'15'
D****************************************************************************
D* Define break
DBreak            C                   '<BR>'
```

Figure 155 (Part 2 of 5). ILE RPG Program ORDAS400G Method=GET

```
     ************************************************************
B
     * Get the Environment Variable called QUERY_STRING
     * Set the ENVarName to QUERY_STRING
     *You must count the length of the Var Name!
     *Set the ENVarLn to this length (12 In This Case)
    C                   MOVEL     *BLANKS       ENBuff
    C                   MOVEL     'QUERY_STRING'ENVarName
    ****************************************************************
    C                   Z-ADD     12            ENVarLn
    ****************************************************************
    C                   EXSR      GETENV
     *  Upon return, your Query_String data is in ENBuff with the len
     *  of the data returned in ENActLn
     *  Move this data to the DBCvt parms
     ******************************************************************************
    C                   Z-ADD     103           DBDSLn
     ******************************************************************************
    C                   MOVEL     ENBuff        DBBuff
    C                   Z-ADD     ENActLn       DBBuffLn
B
     ******************************************************************************
     * Circumvention for HIDDEN fields  ( find out MEMBER-NAME for HTMLO )
     ******************************************************************************
    C                   MOVEL     ENBuff        M14             14
    C                   MOVE      M14           M10             10
    C     '+':' '       XLATE     M10           MBR             10
     ********************************************************************
     * END Circumvention
     ********************************************************************
     * Parse using the CvtDB API
    C                   EXSR      PARSE
     * The field names in your Ext DS now
     * contain  the Values passed in the POST data
     * Move them to the DB file fields
     ******************************************************************************
     * OVR HTMLOUT with MEMBER ORDAS400 and open file
    C                   MOVEA     OVRDBF(1)     OVRARR
    C                   MOVEA     MBR           OVRARR(24)
    C                   MOVEA     OVRARR        CMD             80
    C                   Z-ADD     80            L               15 5
    C                   CALL      'QCMDEXC'
    C                   PARM                    CMD
    C                   PARM                    L
    C                   open      HTMLO
     ********************************************************************
     * Move FORM Input to Database fields
     ******************************************************************************
    C                   MOVEL     NAME          NAME_X
    C                   MOVEL     ADDRESS       ADDRESS_X
    C                   MOVEL     TYPE          TYPE_X
    C                   MOVEL     SERV          SERV_X
     ********************************************************************
     * Write Database record file ORDASOOF
     ******************************************************************************
    C                   WRITE     ORDASR
     * If you had multiple values for the same field, you would
     * have lost all but the first. You need another technique for
     * this situation
```

Figure 155 (Part 3 of 5). ILE RPG Program ORDAS400G Method=GET

```
 ****************************************************************************
 * Create the HTML Output
 * Write HTML Required control records
 * ADD NewLine append after 80 to 120 characters to OutBuff
 ****************************************************************************
C                   DO        9             I                5 0
C                   READ      HTMLOUT                                  98
C     OutBuff       cat       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
 * Prepare variable OUTPUT
C     I             ifeq      9
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       cat       NAME:0        OutBuff
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       cat       ADDRESS:0     OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C     TYPE          ifeq      'P1'
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       CAT       Break:0       OutBuff
C     10            CHAIN     HTMLOUT                          98
C     OutBuff       CAT       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C                   endif
C     TYPE          ifeq      'P2'
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       CAT       Break:0       OutBuff
C     11            CHAIN     HTMLOUT                          98
C     OutBuff       CAT       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C                   endif
C     TYPE          ifeq      'P3'
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       CAT       Break:0       OutBuff
C     12            CHAIN     HTMLOUT                          98
C     OutBuff       CAT       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C                   endif
C     OutBuff       CAT       Break:0       OutBuff
C     OutBuff       CAT       Break:0       OutBuff
C     SERV          ifeq      'T'
C     13            CHAIN     HTMLOUT                          98
C     OutBuff       CAT       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C                   else
C     14            CHAIN     HTMLOUT                          98
C     OutBuff       CAT       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C                   endif
C                   endif
C                   ENDDO
C                   MOVE      *OFF          *IN98
C     15            setll     htmlout
C     *IN98         DOWEQ     *OFF
C                   read      htmlout                              98
C     *IN98         CABEQ     *ON           EndHTM
C     OutBuff       cat       SRCDTA:0      OutBuff
C     OutBuff       CAT       NewLine:0     OutBuff
C     EndHTM        TAG
 * End variable OUTPUT
C                   ENDDO
```

Figure 155 (Part 4 of 5). ILE RPG Program ORDAS400G Method=GET

```
      * Read HTMLFILE until EOF
     C                   MOVE      *OFF          *IN99
     C     *IN99         DOWEQ     *OFF
     C                   READ      HTMLREC                               99
     C                   ENDDO
      * Send OutBuff to standard output
     C     OutBuff       CAT       NewLine:0     OutBuff
     c     ' '           CHECKR    OutBuff       OutBuffLn
     C                   EXSR      STDOUT
      * End program
     C                   CLOSE     HTMLO
     C                   MOVEA     OVRDBF(2)     OVRARR
     C                   MOVEA     OVRARR        CMD             80
     C                   CALL      'QCMDEXC'
     C                   PARM                    CMD
     C                   PARM                    L
     C                   MOVE      *ON           *INLR
      * These are the APIs used in subroutines to keep the main processing
      * simple.  They do not need to be SUBRs!
C
     * Subroutine to Get Environment Variable
     C     GETENV        BEGSR
     C                   CALLB     APIEnVar
     C                   parm                    ENBuff
     C                   parm                    ENBuffLn
     C                   parm                    ENActLn
     C                   parm                    ENVarName
     C                   parm                    ENVarLn
     C                   parm                    QUSEC
     C                   ENDSR
C
     C*  Parse subroutine
     C     PARSE         BEGSR
     C                   CALLB     APICvtDB
     C                   parm                    DBFileName
     C                   parm                    DBBuff
     C                   parm                    DBBuffLn
     ** Remember to code your External DS name.  The API returns your data
     ** in this structure. The field names are in the structure
      ************************************************************************
     C                   parm                    ORDASDS
      ************************************************************************
     C                   parm                    DBDSLn
     C                   parm                    DBActLn
     C                   parm                    DBRespCd
     C                   parm                    QUSEC
     C                   ENDSR
      * This is the STD OUT SUBR
     C     STDOUT        BEGSR
     C                   callb     APIStdOut
     C                   parm                    OUTBuff
     C                   parm                    OUTBuffLn
     C                   parm                    QUSEC
     C                   ENDSR
**CTDATA OVRDBF
OVRDBF FILE(HTMLO) MBR(1234567890) LVLCHK(*NO) OVRSCOPE(*JOB)
DLTOVR FILE(HTMLO) LVL(*JOB)
```

Figure 155 (Part 5 of 5). ILE RPG Program ORDAS400G Method=GET

9.1.6.2 Source-Code RPG Program ORDAS400P

ILE RPG program example for the Method=POST: The logic of this program is identical with the program in Figure 155 on page 209.

Only three parts are replaced:

A by **1**,
B by **2** and
C by **3**

This CGI program receives the name/value pairs from the Form as encoded form input from STDIN.

The environment variable CONTENT_LENGTH determines how much data **must** be read from STDIN.

The following example shows the Read from STDIN QtmhRdStin (subroutine STDIN).

1

```
     ****************************************************************************
     *Variables for the CGI interface APIs
     *These are used for APIStdIn
    DInData           S           2048A   INZ
    DInDataLn         S              9B 0 INZ(2048)
    DInActLn          S              9B 0
     ****************************************************************************
```

2

```
     ****************************************************************************
     * Get the Input parameters from the POST from STDIN
    C                   MOVE      *BLANKS       OutBuff
    C                   EXSR      STDIN
     *  Upon return, your POST data is in INData and its length is in
     *  INActLn  It is in the FLD=VAR format at this time
     *  Move this data to the DBCvt parms
     * Set up the parameters before CALLB
     *  This includes the length of your Ext DS (103 is correct)
     ****************************************************************************
    C                   Z-ADD     103           DBDSLn
     ****************************************************************************
    C                   MOVEL     INData        DBBuff
    C                   Z-ADD     INActLn       DBBuffLn
     ****************************************************************************
```

3

```
     ****************************************************************************
     * Subroutine to read STD IN
    C     STDIN         BEGSR
    C                   CALLB     APIStdIn
    C                   parm                    INData
    C                   parm                    INDataLn
    C                   parm                    INActLn
    C                   parm                    QUSEC
    C                   ENDSR
     ****************************************************************************
```

Figure 156. ILE RPG Program ORDAS400P Method=POST

9.1.6.3 Source-Code C Program PARSECGIP

ILE C program example for the Method=POST: This is the program for Ordering an AS/400 system in the C-Language.

C Programs #include qp0z1170.h

The **qp0z1170.h** must included to get the getenv() and putenv() functions which are not part of the **stdio.h** at this time.

The **qp0z1170.h** is a member in the file **H** in the library **QSYSINC**.

```
/*********************************************************************
/*  Simple ILE C program PARSECGIP to test CGI Method=POST           *
/*                                                                   *
/*                                                                   *
/* 1. Compile this source member as module PARSECGIP ( PDM Option=15 )  *
/*                                                                   *
/* 2. Create program PARSECGIP from module PARSECGIP ( PDM Option=26 )  *
/*    with PROMPT(PF4) and BNDSRVPGM(ITSOIC400/UTIL)                 *
/*                                                                   *
/*********************************************************************
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <qp0z1170.h>

#define MAX_ENTRIES 10000

typedef struct {
    char *name;
    char *val;
} entry;

char *makeword(char *line, char stop);
char *fmakeword(FILE *f, char stop, int *len);
char x2c(char *what);
void unescape_url(char *url);
void plustospace(char *str);


main(int argc, char *argv[]) {
    entry entries[MAX_ENTRIES];
    register int x,m=0;
    int cl;

    printf("Content-type: text/html\n\n");

    if(strcmp(getenv("REQUEST_METHOD"),"POST")) {
        printf("This script should be referenced with a METHOD of POST.\n");
        exit(1);
    }
    if(strcmp(getenv("CONTENT_TYPE"),"application/x-www-form-urlencoded")) {
        printf("This script can only be used to decode form results. \n");
        exit(1);
    }
    cl = atoi(getenv("CONTENT_LENGTH"));
    for(x=0;cl && (!feof(stdin));x++) {
        m=x;
        entries[x].val = fmakeword(stdin,'&',&cl);
        plustospace(entries[x].val);
        unescape_url(entries[x].val);
        entries[x].name = makeword(entries[x].val,'=');
    }

    printf("<html>\n");
    printf("<body>\n");
    printf("<H1>Query Results</H1>");
    printf("You submitted the following name/value pairs:<p>");
    printf("<ul>");

    for(x=0; x <= m; x++)
        printf("<li> <code>%s = %s</code>",entries[x].name,
                entries[x].val);
    printf("</ul>");
    printf("</body>\n");
    printf("</html>\n");
}
```

Figure 157. ILE C Program PARSECGIP Method=POST

9.2 DB2 World Wide Web Connection

This section describes the implementation of DB2 World Wide Web (WWW) which is a macro language that makes the access to SQL based DB2 data very easy for the web client.

Important note

If you would like to follow along with the examples in this and other chapters you first will want to install the ITSO Company demonstration and other web based applications from the CD-ROM that came with this redbook. Please see Appendix A, “Installing the ITSO Company Demo” on page 285 for instructions on how to get your AS/400 up and running right away.

9.2.1 An Overview of DB2 World Wide Web Connection

With the fast growing popularity of Internet and World Wide Web, there is also a fast growing demand for Web access to databases. DB2WWW Connection is a Web server gateway to DB2 family in IBM (see Figure 158).

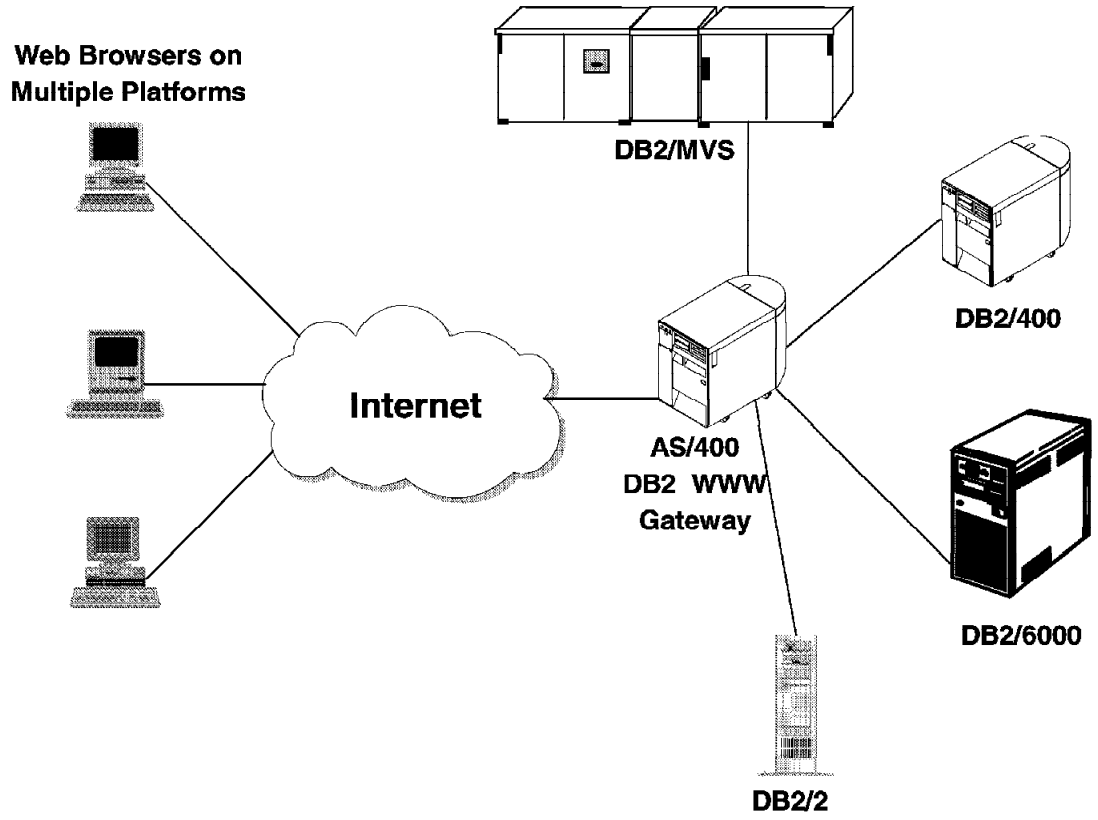


Figure 158. DB2WWW Overview

It enables the application developer to build Web applications for DB2 databases using HTML forms and dynamic SQL. End users of these DB2 Web applications see only the forms for their requests and resulting reports. A user fills out the forms, points and clicks to navigate the forms, and accesses the database as determined by the application. The complete SQL command is dynamically built with user inputs and sent to the database.

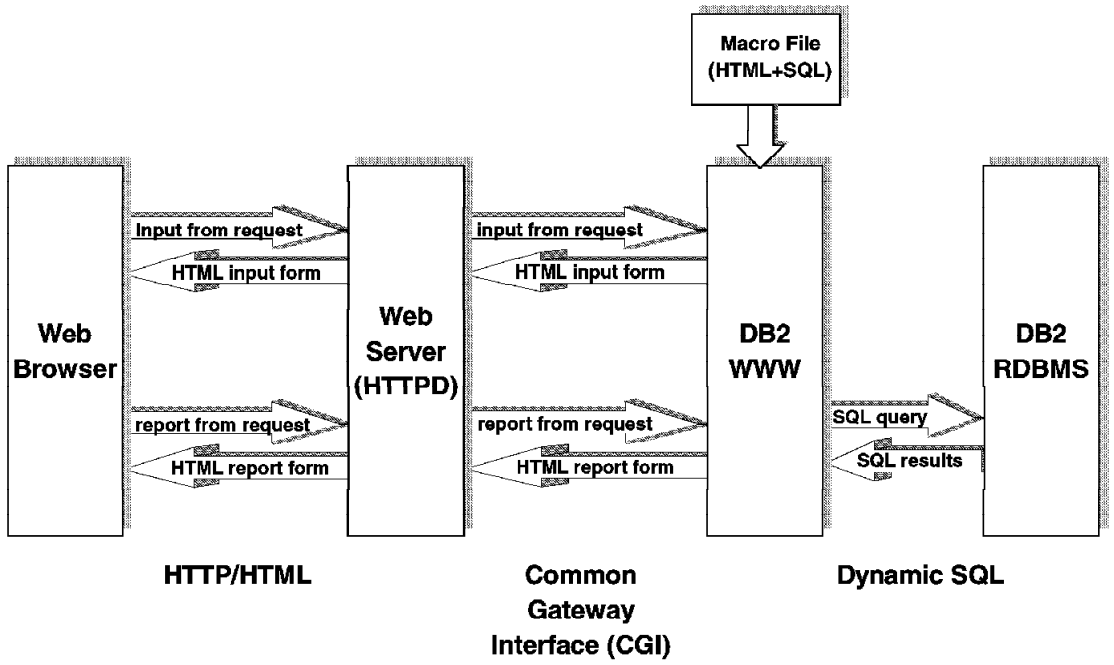


Figure 159. DB2WWW System Overview

The *Cross-language variable substitution* was developed to provide a simple and flexible variable substitution, enabling the application developer to use the full capabilities of HTML for creation of query forms and reports, and SQL for queries and updates. The application developer creates HTML forms and SQL commands, and stores them in macro files at the Web server. Variables are used to link the SQL commands and the HTML forms within the same macro file. These macro files get processed by the DB2WWW Connection runtime engine.

9.2.1.1 Features and Functions

The DB2WWW Connection was designed with the following objectives:

- To not require extensive programming by the application developer other than the use of HTML to create forms and SQL for queries and updates against the database.
- To be sufficiently flexible for a variety of Web applications that do not require extensive programming logic.
- To be portable to multiple server platforms.
- To be usable with existing Web HTML editors and database query tools.

These goals lead to the development of the DB2WWW Connection macro language. The macro language is a combination of HTML, SQL, and a simple *cross-language variable substitution mechanism* that allows:

1. Input data from the HTML input form to be inserted into the SQL calls to the databases.
2. SQL query results to be fed back into HTML report forms.

The DB2WWW Connection runtime engine reads the macro files to generate the appropriate query forms, SQL commands, and report forms. DB2 WWW Connection has the following features:

- DB2WWW Connection applications can use native HTML and SQL, thus exploiting the expressive power of these languages without artificial limitations.
- The DB2WWW Connection runtime gateway is small and efficient, and is readily portable to multiple platforms.
- Visual tools may be used to partially generate DB2WWW Connection macro files. Third party vendors' visual HTML editors can be used to generate the HTML form sections, while various visual query tools such as *IBM's Visualizer Query* may be used to generate the SQL sections.

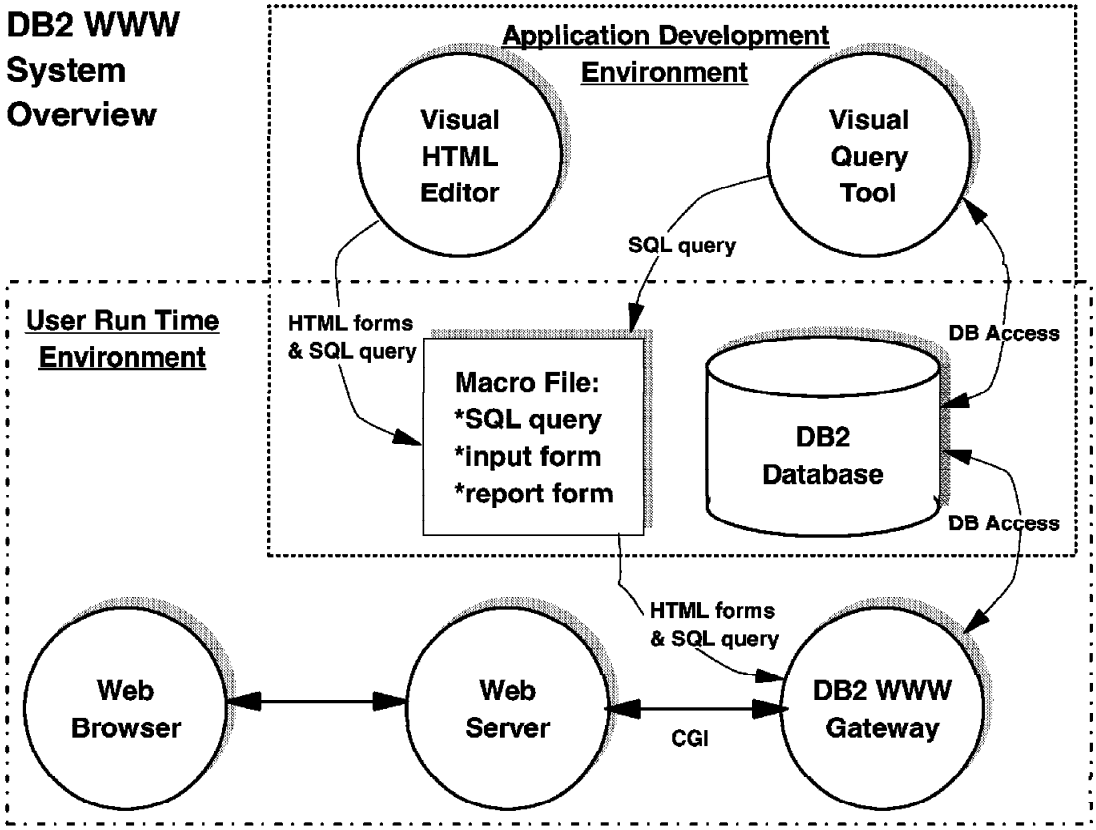


Figure 160. DB2WWW Runtime Control Flow

9.2.1.2 DB2WWW Connection and Internet Security

Please note that the following section is a generic description for the DB2 family, not for the AS/400 system.

The DB2WWW Connection works with the DB2 database, the Web server, and the firewall products to provide secure data access over the Internet. When used with one or more of these other products, the types of security provided are as follows:

- *Authentication:* A DB2WWW Connection gateway has two types of authentication (user login and password), one provided by the Web server and the other by DB2. They are as follows:
  - Web servers can typically be configured to protect certain directories on the server. When a URL accesses a file in that directory, login and password pop-up windows appears on the end user's Web browser to request for authentication.
  - DB2 has a login/password authentication mechanism for database access. This mechanism can be used to restrict access to tables and columns by certain users.
- *Encryption:* DB2WWW Connection, when used with a secure Web server that has support for Secure Sockets Layer (SSL) or Secure HyperText Transfer Protocol (SHTTP), or both, benefits from the public key encryption scheme provided by the secure Web server and secure Web client. The user login and password used for authentication are encrypted for transmission, as well as all user inputs to the forms and the query results that are sent back to the user. A secure Web server/client combination is a must for protection of sensitive data.
- *Firewall:* DB2WWW Connection may be used with firewall products, such as NetSP for IBM, which protects both the DB2WWW Connection machine and the corporate network from external internet attacks. Various configurations and filter settings for this firewall are being tested.

For the most secure access to DB2 data, DB2WWW Connection should be used in conjunction with both a secure Web server and the NetSP firewall.

The SQL section must be defined before the HTML. The macro file gets processed by the DB2WWW Connection runtime engine, resulting in the appropriate fill-in forms for the user to input or select values, the SQL command being generated, and the resulting report being generated.

9.2.2 DB2WWW for AS/400 System

Important Note

Because the implementation of the DB2WWW for the AS/400 system uses commitment control, all files that are accessed for update through it must be journaled. Files that are accessed via the SQL SELECT statement do not need to be journaled.

PTF Information

This section and the macro examples that are included with this redbook assume that you have PTF SF34455 for 5763-TC1 loaded and applied to your system.

The AS/400 HTTP server provides access to DB2 databases in a manner consistent with *DB2WWW* from which it is derived. DB2WWW is an extension of the AS/400 HTTP server. The HTTP server recognizes requests for DB2WWW when parsing the URL of a client's request and passes such requests to the DB2WWW server extension.

The format for a DB2WWW URL is:

```
QSYS.LIB(/library.LIB)/db2www.PGM/{macro-file}/{command}(?name=val&...)
```

The request for DB2WWW is a special program call; the program to be called is *db2www*. The server recognizes a DB2WWW request by the program name (db2www) and passes the request to the *db2www* extension of the server.

The *db2www* extension to the server meets the CGI interface for program calls. It expects the macro file name and command to be passed in an environment variable named ″PATH_INFO″. The remaining parameters are passed either in the ″QUERY_STRING″ environment variable or in the standard input stream depending on the request method being set to **GET** or **POST**. See 9.1, “Common Gateway Interface Programs” on page 191 to better understand these concepts.

The macro file named in the DB2WWW request (HTTP_DB2_MACRO) names a source physical file in the library identified by the LIB parameters of the request URL. The macro file defines how the request is to be processed and is consistent with the DB2WWW specification. It has four sections:

- Variable definition section
- SQL command section
- HTML input section
- HTML output section

The SQL query section control of the macro file defines the SQL command to be processed against the DB2 database when the DB2WWW is set to *report*.

The query input form section of the macro file provides the HTML that is presented to the client browser when the DB2WWW request command is set to *input*. The HTML in the input section is intended to allow the user at the client browser to provide inputs required for the SQL query. The HTML document provided by the input section should be set up so that when the client completes providing the query inputs, the next DB2WWW request command to the server is *report* which causes the server to query the database and produce a query report.

The query report section of the macro file defines the format of the document to be produced when the DB2WWW request command is *report*. If no report section is provided in the macro file, a default report is generated.

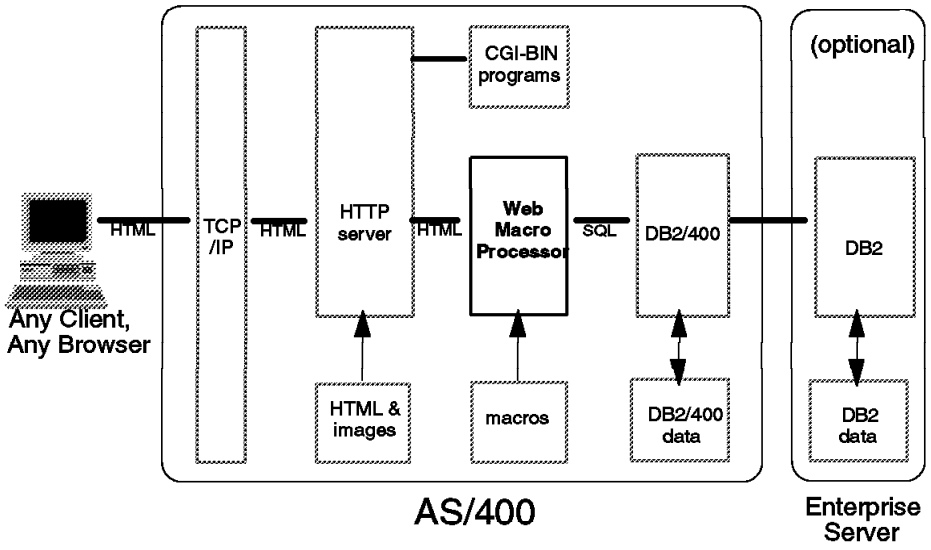


Figure 161. Web Macro Server Structure

Figure 161 shows the structure of Macro Processor. As you can see, when the user requests the URL, the HTTP server passes the URL to the Macro Processor. Then it processes the HTML input section in macro file. After the user inputs related data and clicks the submit button, the Macro Processor executes the SQL command section to fetch data from DB2/400 and then processes the HTML output section to display the result.

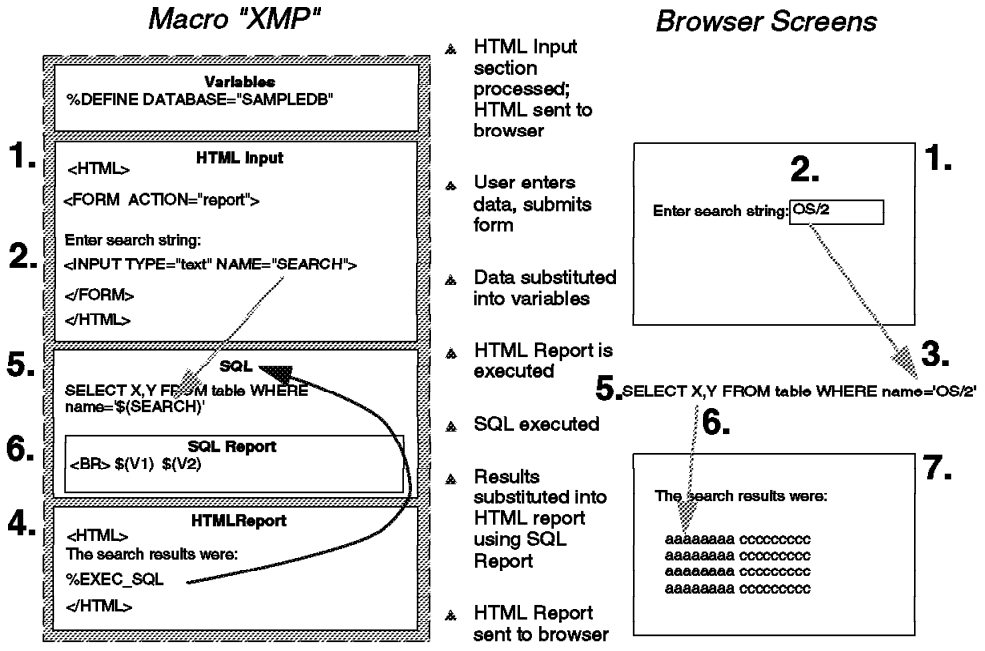


Figure 162. Web Macro Functional Overview

Figure 162 is the functional overview of Macro File. You can see the format of macro file on the left side and the sequence of the entire process in the middle.

9.2.2.1 Build Your DB2WWW Function in AS/400 System

It is quite easy to build the DB2WWW function in AS/400 system. Just use the following the steps:

1. Create a library (for example, ITSOIC400). This is for your DB2WWW applications.
2. Create a source physical file named INI in library ITSOIC400 to be the DB2WWW programs initialization file.
3. Create a member named DB2WWW in INI source physical file.
4. Edit DB2WWW member with two statements.

   ```
   MACRO_PATH = /HTML
   CONVERT_CODE_PAGE = NO
   ```

   The MACRO_PATH statement tells the DB2WWW program in which file the macros are.
5. Create a source physical file named HTML in library ITSOIC400.
6. Create your macro member in HTML file. (You can use the sample program in the following section.)
7. Add the following directive into HTTP configuration file (unless you already have it in there):

   ```
   Exec /QSYS.LIB/QTCP.LIB/*
   ```

   This directive allows the DB2WWW program to be run. It also allows *any* program in the QTCP library to be called by the HTTP server.

   Note: The CERN Exec directive specifies the path *to* the program to be executed, not the actual program. So, an HTTP directive like Exec /QSYS.LIB/QTCP.LIB/DB2WWW.PGM would not be valid.
8. Add the following directives into HTTP configuration file:

   ```
   MAP /cgi-bin/db2www/* /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/*
   MAP /CGI-BIN/DB2WWW/* /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/*
   ```

   These directives map DB2WWW requests to the DB2WWW program. We include both an upper and lower case for convenience.
9. Add the following directives into HTTP configuration file:

   ```
   Pass /qsys.lib/itsoic400.lib/html.file/*
   Pass /QSYS.LIB/ITSOIC400.LIB/HTML.FILE/*
   ```

   These directives allow the DB2WWW macros to be accessed. We include both an upper and lower case for convenience.

That is all you need to do. Very simple!

Then from Web Browser, open the URL

```
http://your-hostname/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/your-homepage.MBR
```

to start the DB2WWW journey.

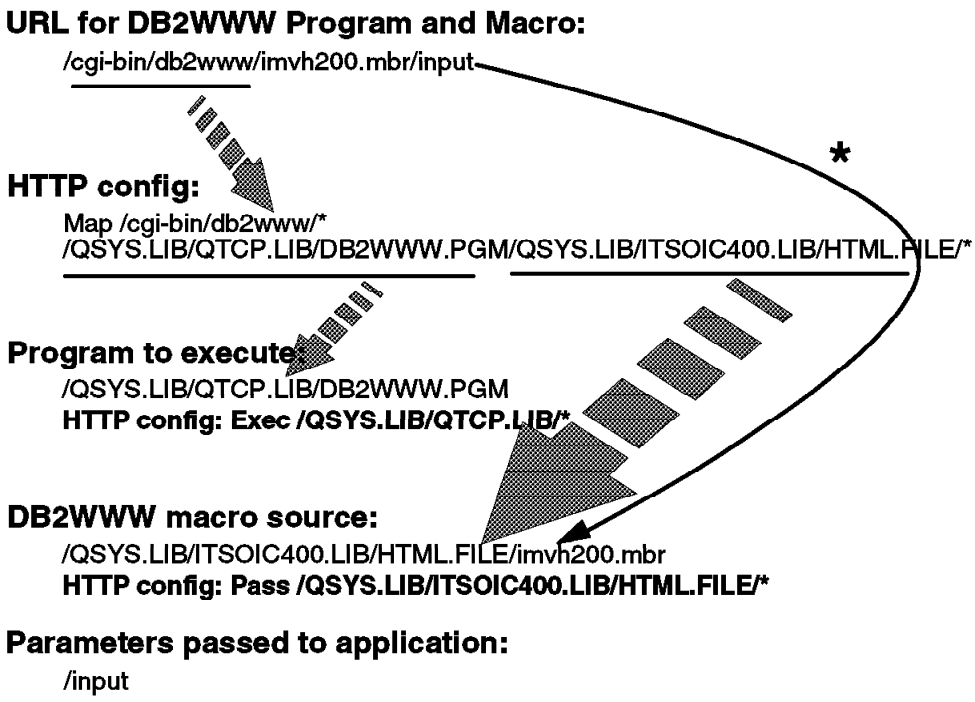


Figure 163. Mapping DB2WWW URL to HTTP Configuration

9.2.3 Examples of DB2WWW for AS/400 System

The following source files are in the sample library ITSOIC400.

1. INI source physical file with member DB2WWW.
2. HTML source physical file with member HOME, SIMPLE, and COMPLEX.

After installing the sample library ITSOIC400 and directory /ITSOIC.400, you need to make sure the following four directives are in the HTTP configuration file. If you ran the installation program that came with this book (please see Appendix A, “Installing the ITSO Company Demo” on page 285) then these statements should have already been added for you. If you already have added your own map and pass directives for the DB2WWW, you need to delete or comment them out.

```
MAP /cgi-bin/db2www/* /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/*
MAP /CGI-BIN/DB2WWW/* /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/*
PASS /QSYS.LIB/ITSOIC400.LIB/HTML.FILE/*
PASS /qsys.lib/itsoic400.lib/html.file/*
```

After making these changes to the HTTP configuration, restart the server by using the STRTCPSVR SERVER(*HTTP) RESTART(*HTTP) command or if the server is not running, then start it by using the STRTCPSVR SERVER(*HTTP) command. From a web browser, open the URL http://hostname/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/HOME.MBR and then you can test the DB2WWW functions.

The first sample program is the home page you see when you open the URL. Selecting Simple or Complex Test leads you to the next two sample macro file programs.

Figure 164 on page 224 shows the display you see when you open the URL on your web browser and Figure 165 on page 224 is the HTML file HOME in source physical file HTML in ITSOIC400 library.

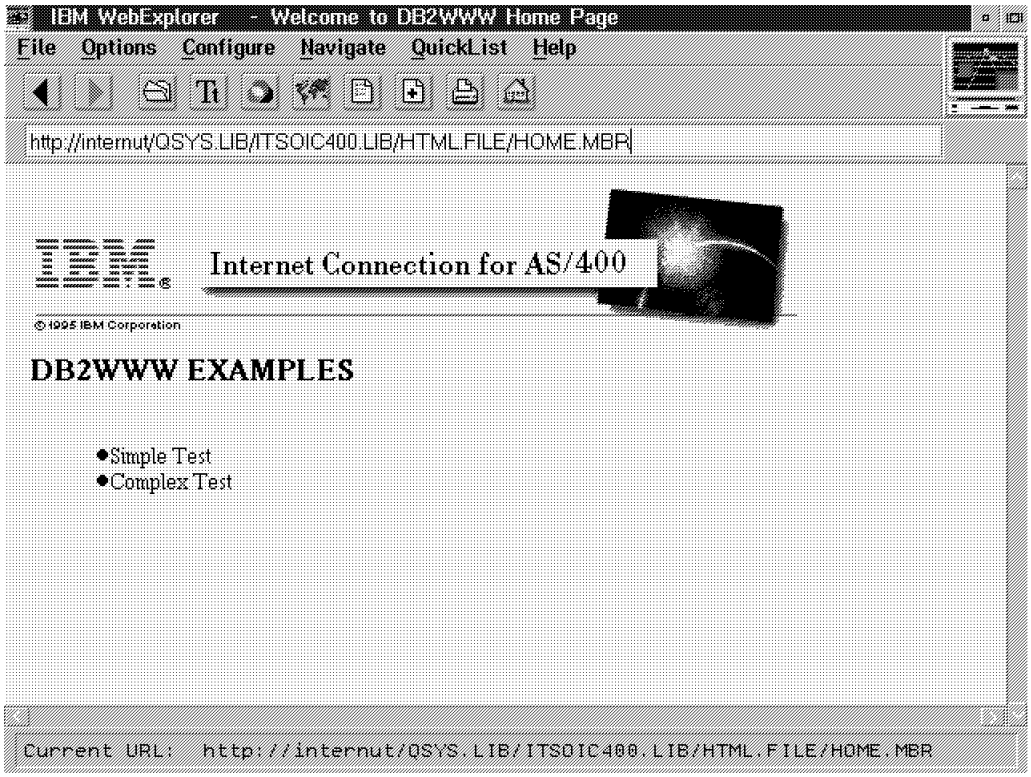


Figure 164. Display of Sample Program - Home Page

```
<!                                                                  ->
<! Below are all HTML tags                                          ->
<! Pay attention to the HREF statement.                             ->
<! (1) /cgi-bin/db2www/ means the path where db2www macro processor ->
<!     is.  As a matter of fact in HTTP configuration file there is ->
<!     a MAP and PASS statement for this path.                       ->
<! (2) simple.mbr is the macro file you create.                     ->
<! (3) complex.mbr is the macro file you create.                    ->
<! (4) input means execute the macro file HTML_INPUT section        ->
<!                                                                  ->
<HTML>
<HEAD>
<TITLE>Welcome to DB2WWW Home Page</TITLE>
</HEAD>
<BODY BGCOLOR="#F8F8FF">
<IMG SRC="/BonusImg/ICAS400.GIF">
<H2>DB2WWW EXAMPLES</H2>
<UL>
<LI><A HREF="/cgi-bin/db2www/simple.mbr/input">Simple Test</A>
<LI><A HREF="/cgi-bin/db2www/complex.mbr/input">Complex Test</A>
</UL>
</BODY>
</HTML>
```

Figure 165. Sample Program - Home HTML

After selecting the Simple Test in Figure 164, the DB2WWW macro processor executes the HTML input section that you see in Figure 166 on page 225.

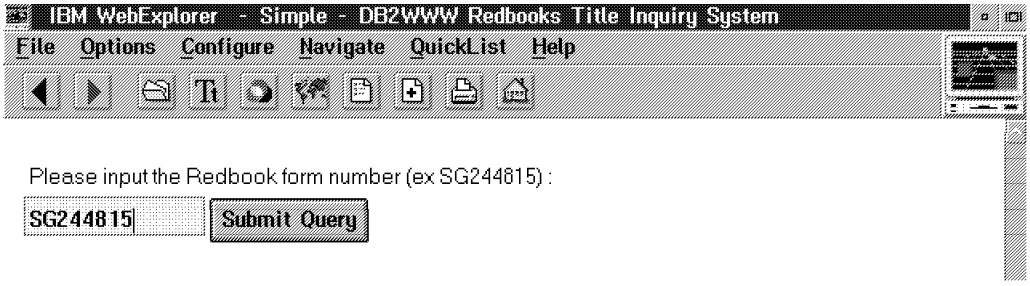


Figure 166. Display of Sample Program - Simple Test Input Window

When you input the form number and click on Submit Query on Figure 166, the DB2WWW macro processor executes the SQL Command Section and the HTML Output Section that you see in Figure 167.

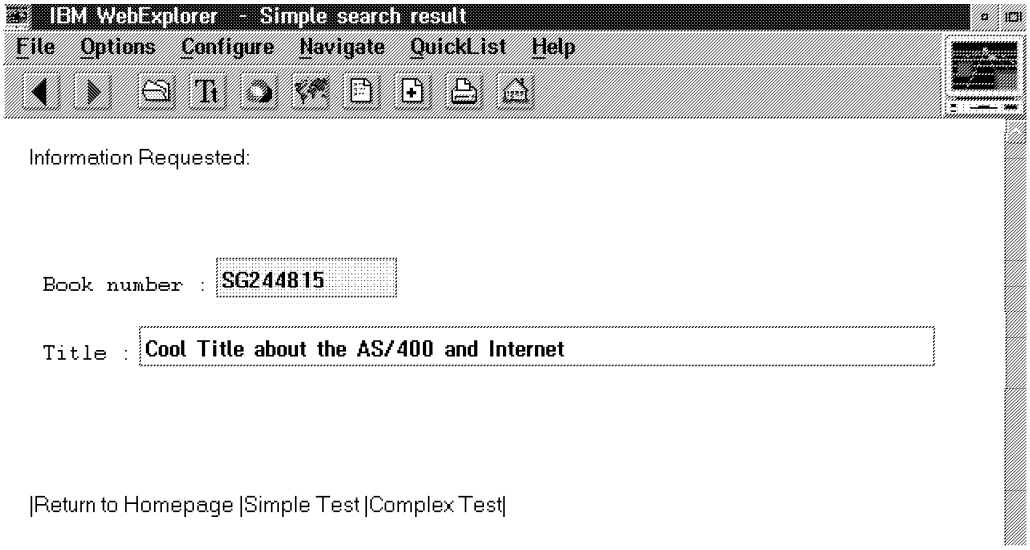


Figure 167. Display of Sample Program - Simple Test Output Window

Figure 168 on page 226 is the macro file source named SIMPLE in the HTML source physical file.

```
%{
This is the comment format, you can use to add your comment in the
macro file.
%}


%{-------------(1) Variable Definition Section-----------------------
%}


%{
This part is the Variable Definition Section.
%}

%{
Define two variables below. One is varl for library name and varf is
for file name used in SQL SELECT sentence.
%}

%define varl="ITSOIC400"
%define varf="REDBOOKS"



%{-------------(2) SQL Command Section-------------------------------
%}


%{
This part is the SQL Command Section.
%}


%{
Here we select two fields from file REDBOOKS in library ITSOIC400 by
using the where condition statement.
    What you must know....
    (1) $(varl) means the value of the variable varl.
    (2) '$(enum)' means the type of the variable enum is character.
    (3) SQL_REPORT{..... is used to format SQL result.
    (4) $(V1) means first variable field in select statement.
    (5) In SQL statement file in library is presented as
        library.file you can see in the from sentence.
    (6) You can set the field size smaller or larger
        than definition in the file.
%}

%SQL{
select  redform,redtitle
from $(varl).$(varf)
where redform = '$(enum)'
%SQL_REPORT{
%ROW{
Book number : <INPUT TYPE="text" SIZE="8" NAME="redform" VALUE="$(V1)">
<br>
Title : <INPUT TYPE="text" SIZE="35" NAME="redtitle" VALUE="$(V2)">
<br>
%}
%}
%}
```

Figure 168 (Part 1 of 3). Sample Program - Simple Macro File

```
%{-------------(3) HTML Input Section---------------------------------
%}


%{
This part is the HTML Input Section.
%}

%{
Below are all HTML tags except for (%HTML_INPUT{).
%}

%{
What you must know is the ACTION part when the user click submit
button, db2www macro processor will execute the SQL statement and
pass back the result to the (%HTML_OUTPUT{) section.

(1) The path name /cgi-bin/db2www/is where db2www macro processor
    reside.
(2) SIMPLE.MBR means the name of this macro file.
(3) The final portion 'report' means to generate output as design
    in the SQL and Output section.
%}

%HTML_INPUT{
<HTML>
<HEAD>
<TITLE>Simple - DB2WWW Redbooks Title Inquiry System</TITLE>
</HEAD>
<BODY>
<FORM METHOD="post"
ACTION="/cgi-bin/db2www/SIMPLE.MBR/report">
Please input the Redbook form number (ex SG240372) :<br><br>
<INPUT TYPE="TEXT" NAME="enum" SIZE="8">
<INPUT TYPE="submit" Value="Submit Query"><br>
</FORM>
</BODY>
</HTML>
%}
```

Figure 168 (Part 2 of 3). Sample Program - Simple Macro File

```
%{-------------(4) HTML Output Section--------------------------------
%}


%{
This part is the HTML Output Section
%}

%{
You can format your HTML output here with HTML tags.
(%EXEC_SQL) means the SQL output will be put in here.
%}

%HTML_REPORT{
<HTML>
<HEAD>
<TITLE>Simple search result</TITLE>
</HEAD>
Information Requested:<br>
<PRE>
%EXEC_SQL
</PRE>
<A HREF="/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/HOME.MBR">
|Return to Homepage</A>
<A HREF="/cgi-bin/db2www/SIMPLE.MBR/input">|Simple Test</A>
<A HREF="/cgi-bin/db2www/COMPLEX.MBR/input">|Complex Test|</A>
</HTML>
%}
```

Figure 168 (Part 3 of 3). Sample Program - Simple Macro File

When you click Complex Test on Figure 164 on page 224, the DB2WWW macro processor executes the Complex Test HTML input section that you see in Figure 169.

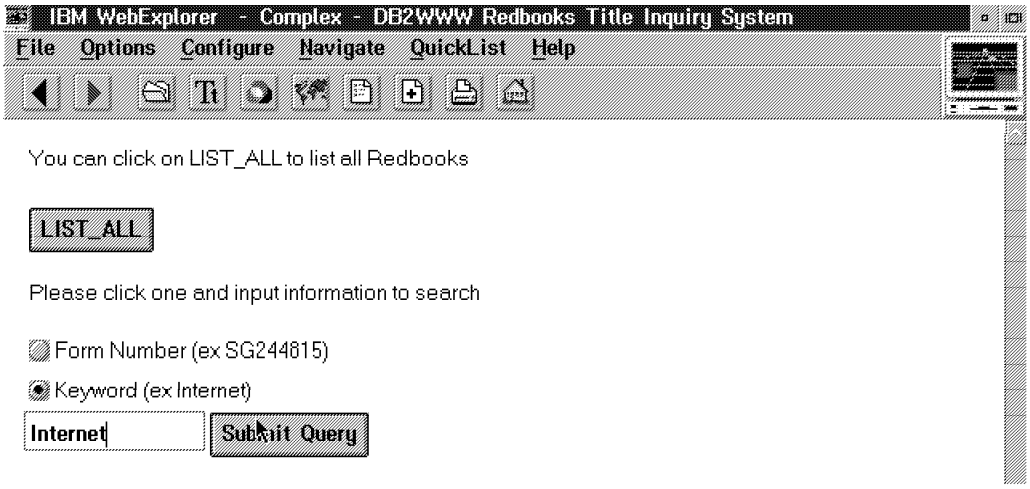


Figure 169. Display of Sample Program - Complex Test Input Window

Figure 170 on page 229 is the window that is shown after you click on the radio button keyword and type **Internet** for search. After you click on Submit Query, then the DB2WWW macro processor executes the SQL command section and the HTML output section.

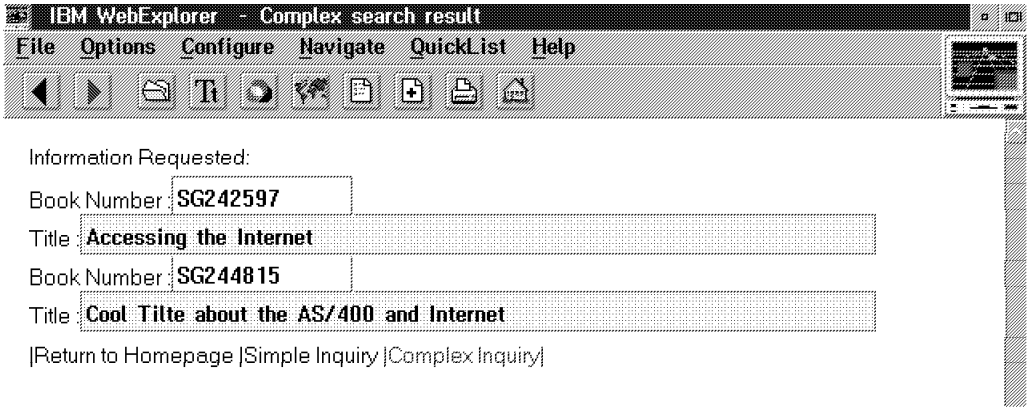


Figure 170. Display of Sample Program - Complex Test Output Window

Figure 171 shows the macro source member named COMPLEX in the source file HTML.

```
%{
Please refer SIMPLE macro file for structure explanation.
Here only explain multiple SQL statement operation.
%}


%{-------------(1) Variable Definition Section-------------------------
%}


%define tl="itsoic400"
%define tf="redbooks"
```

Figure 171 (Part 1 of 3). Sample Program - Complex Macro File

```
%{-------------(2) SQL Command Section---------------------------------
%}


%{
You can see here we have three SQL sections but only one will
executed which depends on user's choice is (1) list all redbooks
(2) inquiry by form number (3) by title keyword.
Corresponding SQL statement declared in the HTML_INPUT section
%}


%{-----------------Multiple SQL----------------------------------------
%}


%SQL(LIST_ALL){  1
select  redform,redtitle
from $(tl).$(tf)
where redform > 'GG000000'
%SQL_REPORT{
%ROW{
Book number : <INPUT TYPE="text" SIZE="8" NAME="redform" VALUE="$(V1)">
<br>
Title : <INPUT TYPE="text" SIZE="35" NAME="redtitle" VALUE="$(V2)">
<br>
%}
%}
%}


%SQL(FORM_NUM){  1
select  redform,redtitle
from $(tl).$(tf)
where redform = '$(enum)'
%SQL_REPORT{
%ROW{
Book number : <INPUT TYPE="text" SIZE="8" NAME="redform" VALUE="$(V1)">
<br>
Title : <INPUT TYPE="text" SIZE="35" NAME="redtitle" VALUE="$(V2)">
<br>
%}
%}
%}


%SQL(KEY_WORD){  1
select  redform,redtitle
from $(tl).$(tf)
where redtitle like '%$(enum)%'
%SQL_REPORT{
%ROW{
Book Number : <INPUT TYPE="text" SIZE="8" NAME="redform" VALUE="$(V1)">
<br>
Title : <INPUT TYPE="text" SIZE="40" NAME="redtitle" VALUE="$(V2)"><br>
%}
%}
%}
```

Figure 171 (Part 2 of 3). Sample Program - Complex Macro File

```
%{-----------------HTML Input section-----------------------------------
%}


%{
(1) Here we have two forms user can choose.
(2) The trick in multiple SQL is the 'name'.  You can see in each input
    tag has the name="A" and with different value.  1
(3) When user choose one way for inquiry then in the HTML_OUTPUT
    section %EXEC_SQL($(A)) will substitute with corresponding value
    into and execute that SQL
%}


%HTML_INPUT{
<HTML>
<HEAD>
<TITLE>Complex - DB2WWW Redbooks Title Inquiry System</TITLE>
</HEAD>
<BODY>
<! list all redbooks                                                >
You can click on LIST_ALL to list all Redbooks<br>
<FORM METHOD="post"
ACTION="/cgi-bin/db2www/COMPLEX.MBR/report">
<input type="submit"  value="LIST_ALL" name="A"><br>
</FORM>
<! search by form number or keyword                                 >
Please click one and input information to search<br>
<FORM METHOD="post"
ACTION="/cgi-bin/db2www/COMPLEX.MBR/report">
<INPUT TYPE="radio" name="A" value="FORM_NUM">Form Number
(ex SG240372)<br>
<INPUT TYPE="radio" name="A" value="KEY_WORD">Keyword (ex Internet)<br>
<INPUT TYPE="TEXT" NAME="enum" SIZE="8">
<INPUT TYPE='submit' Value='Submit Query'><br>
</FORM>
</BODY>
</HTML>
%}


%{-----------------HTML Output Section----------------------------------
%}


%HTML_REPORT{
<HTML>
<HEAD>
<TITLE>Complex search result</TITLE>
</HEAD>
Information Requested:<br><br>
%EXEC_SQL($(A))  1
<br>
<A HREF="/QSYS.LIB/ITSOIC400.LIB/HTML.FILE/HOME.MBR">
|Return to Homepage</A>
<A HREF="/cgi-bin/db2www/SIMPLE.MBR/input">|Simple Inquiry</A>
<A HREF="/cgi-bin/db2www/COMPLEX.MBR/input">|Complex Inquiry|</A>
%}
```

Figure 171 (Part 3 of 3). Sample Program - Complex Macro File

9.2.3.1 Sophisticated Demo Program

When you install the sample program library ITSOIC400, directory ITSOIC.400, and, of course, the related settings, then you can have a look at the "sophisticated demo program" of DB2WWW. Please, just follow the steps:

- From Web Browser, open the URL http://hostname/ITSOIC.400/.

  You see Figure 172 on page 232.

- Click on Sales (Visit our online catalog).

  You see Figure 173 on page 233. The source for this macro file is named IMVH200 in HTML source physical file in library ITSOIC400.

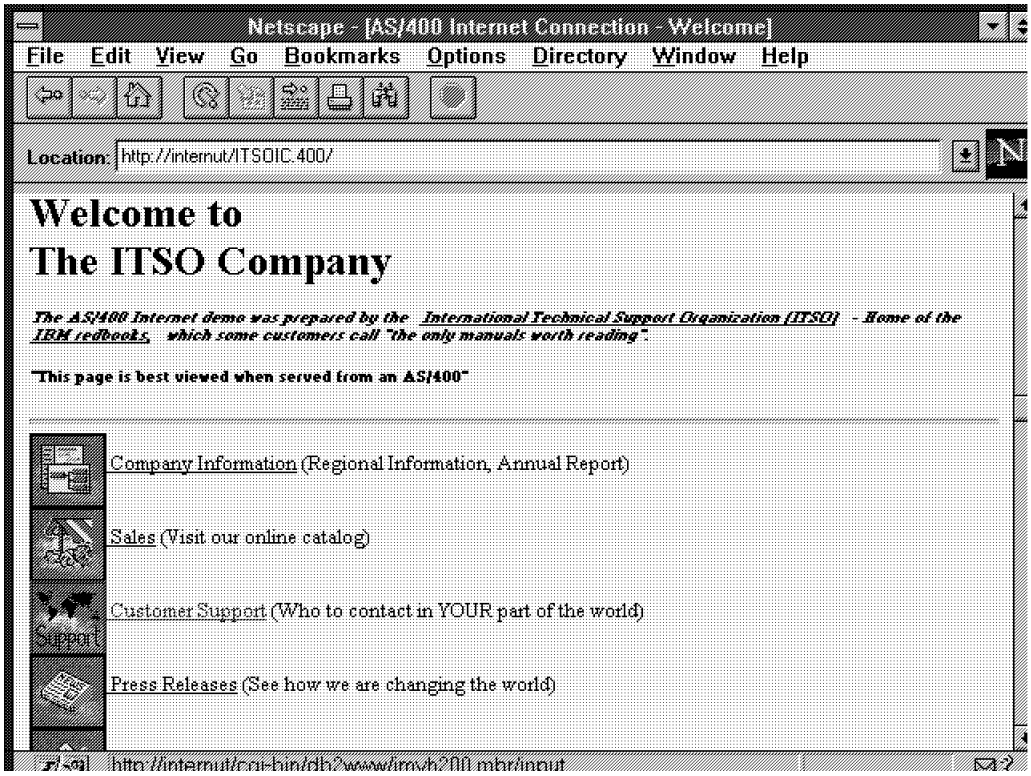


Figure 172. Home Page of Demo Program

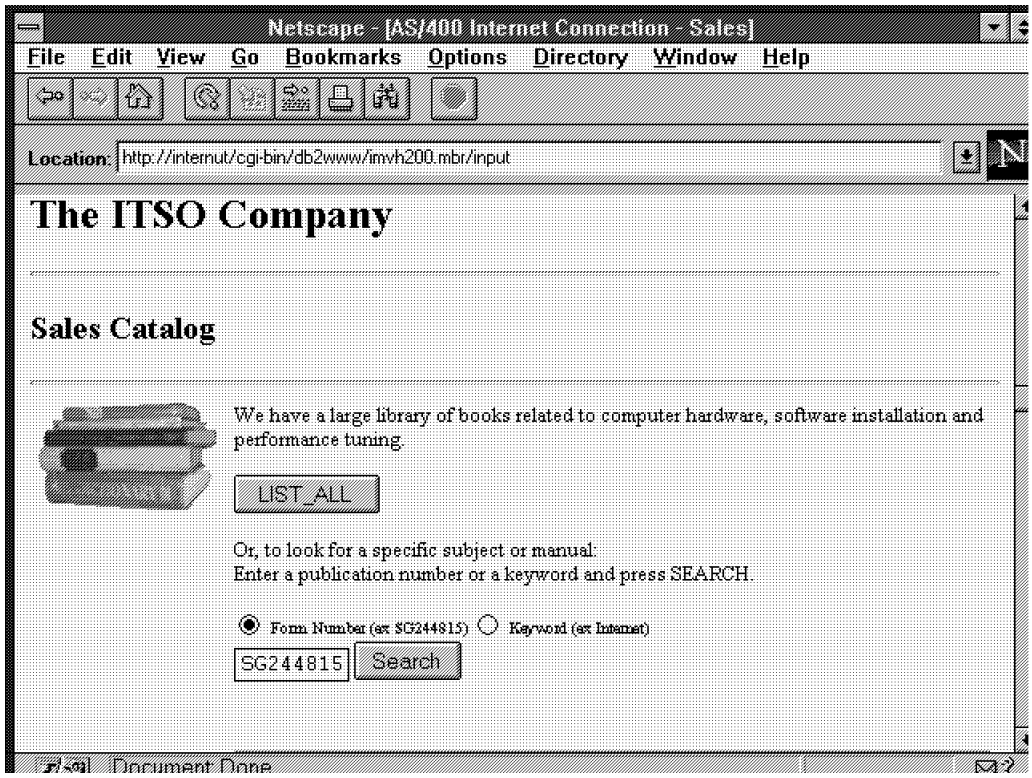


Figure 173. Sales Catalog of Demo Program

9.3 5250 HTML Work Station Gateway

Important note

If you would like to follow along with the examples in this and other chapters you first will want to install the ITSO Company demonstration and other web based applications from the CD-ROM that came with this redbook. Please see Appendix A, “Installing the ITSO Company Demo” on page 285 for instructions on how to get your AS/400 up and running right away.

Important Note

Install PTF SF32373 (for 5763-TC1) before using the 5250 HTML Gateway.

Most Web servers today require that you write scripts or programs to create interactive forms and applications for the World Wide Web. For most software providers, this can mean learning new tools and procedures if they want to support the World Wide Web. This is not true for AS/400 customers. With the AS/400 HTML Gateway function in Internet Connection for AS/400, your current development tools work for creating WWW applications. Once your WWW applications are created, you can start using the Internet′s worldwide reach to open new marketing opportunities. Even existing AS/400 applications can run over the Web without modifying any code. There is no conversion program to run. Just install and configure Internet Connection for AS/400, and the applications on your AS/400 system are ready to go!

So how does IBM do it?

AS/400 applications are inherently display-oriented. That means, each application creates a series of displays for use in its application. These displays are normally sent out in a 5250 data stream to the workstation or emulator, which shows the text. Internet Connection for AS/400 intercepts this 5250 data stream and converts it to HTML, a language the Web understands. Any Web browser used for accessing the World Wide Web can work with the application.

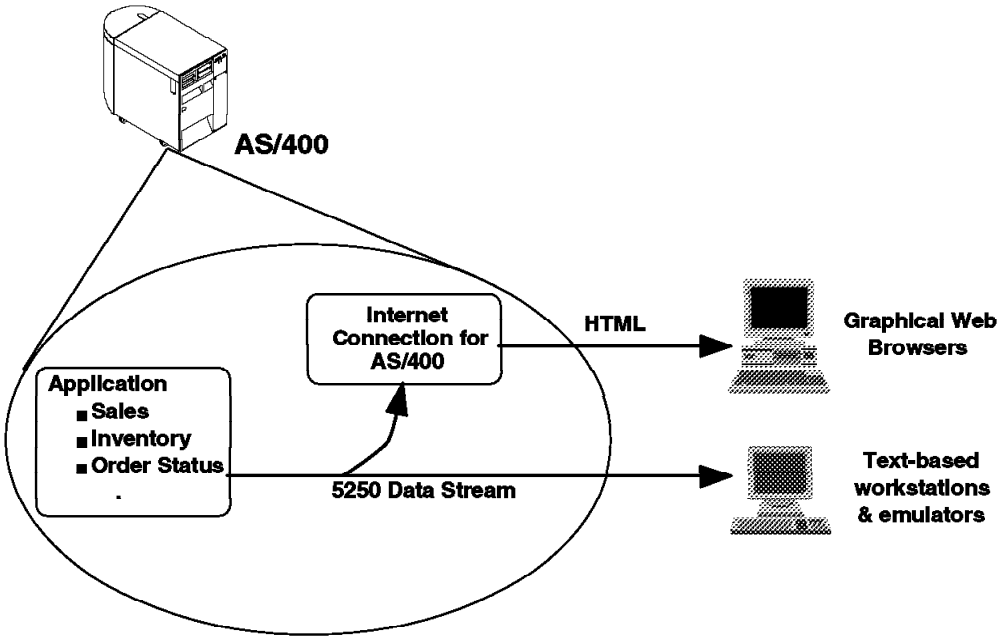


Figure 174. 5250 HTML Gateway

Internet Connection for AS/400 means your business does not need to rely on one specific client platform. Any PC that has a Web browser installed can run AS/400 applications. There is no additional connection configuration. Just point your Web browser to the AS/400 system, and you are in business!

If your business writes AS/400 applications, then Internet Connection for AS/400 means a wealth of new applications on the Internet. You do not need to retrain your programmers. They can continue using their existing development tools (RPG, COBOL, and DDS). Also, with AS/400 HTML Gateway in Internet Connection for AS/400, your programmers can jazz up your applications by adding graphics and any other HTML tags. It requires only a small change to the DDS specifications, and it does not affect your workstation users.

Now that we know what a 5250 HTML gateway does, let's see some examples of the translation from text-based 5250 panels to something a web client can see and use. For this, we are going to show you some OS/400 displays that have been translated to HTML.

1. Sign-on:

   Figure 175 on page 236 shows a portion of the traditional AS/400 sign-on display converted now to HTML and displayed on a WebExplorer client. Note the functionality is really no different than with a normal text-based 5250 emulator.

   The URL that your web client needs to specify to evoke the 5250 to HTML Workstation Gateway support looks similar to the following:

```
http://hostname:5061/WSG/QAPP0100?exit-information
```

Where

http: The Workstation Gateway uses the HTTP protocol.

hostname This identifies the system where the request goes. This can be just the host name or the fully-qualified host name with domain.

:5061 5061 is the default well-known port for the Workstation Gateway server. You must specify this port because your web client tries to connect to port 80 by default if you fail to override this.

WSG Means using the HTML Workstation Gateway function. WSG *must* be uppercase.

/QAPP0100 The prefix that indicates that exit point information follows and *must* be uppercase.

?exit_information

Not shown in the preceding example are the optional parameters that can be used to pass information from the client to the Workstation Gateway server running on the AS/400 system. Characters following the /QAPP0100 are interpreted as parameters to be passed to the server job. For the initial connection, these parameters can be a USERID and password used to direct the new client directly to a 5250 application without the need to sign on to the AS/400 system. Later, after the session has been established, what follows after the WSG is information to allow the AS/400 system to route this display to the proper Workstation Gateway server. This is because the AS/400 system must save state while using a protocol such as HTTP that does not save state! Look closely at the bottom of all of the figures in this section for the URL used to save state.

Please see 9.3.4, “5250 HTML Workstation Gateway Application Logon Exit Program” on page 252 for more information about the Workstation Gateway exit program.

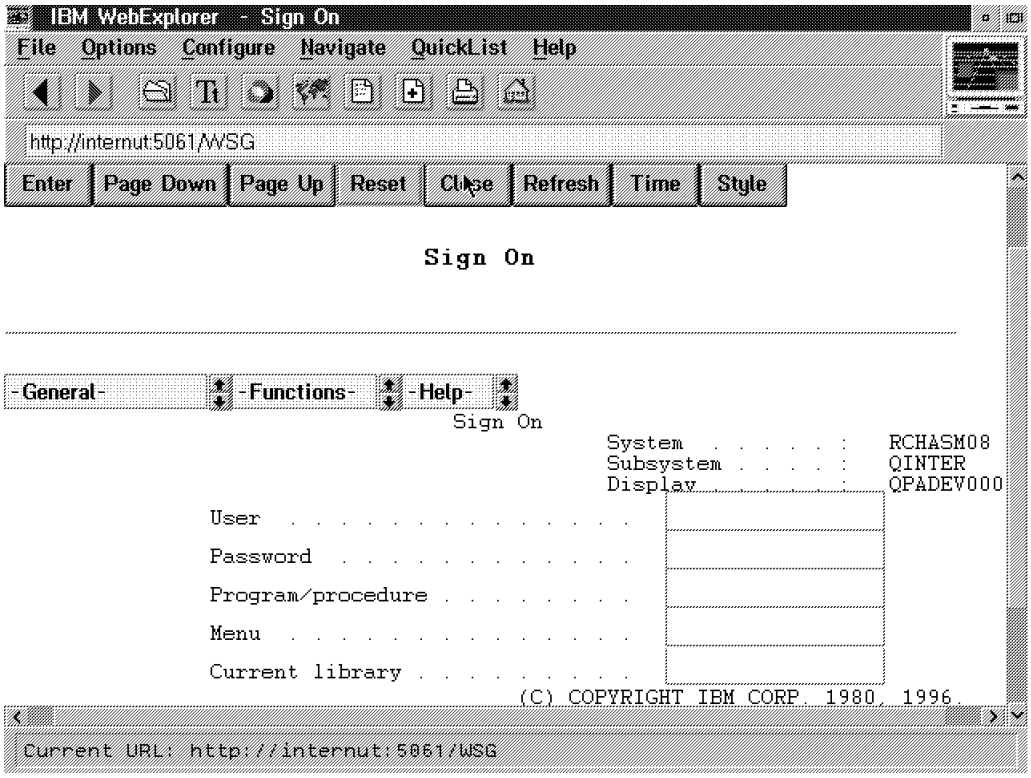


Figure 175. A Portion of the AS/400 Sign-on as seen by the Workstation Gateway

2. Command Entry:

   Figure 176 on page 237 shows the Command Entry display for the WebExplorer client.

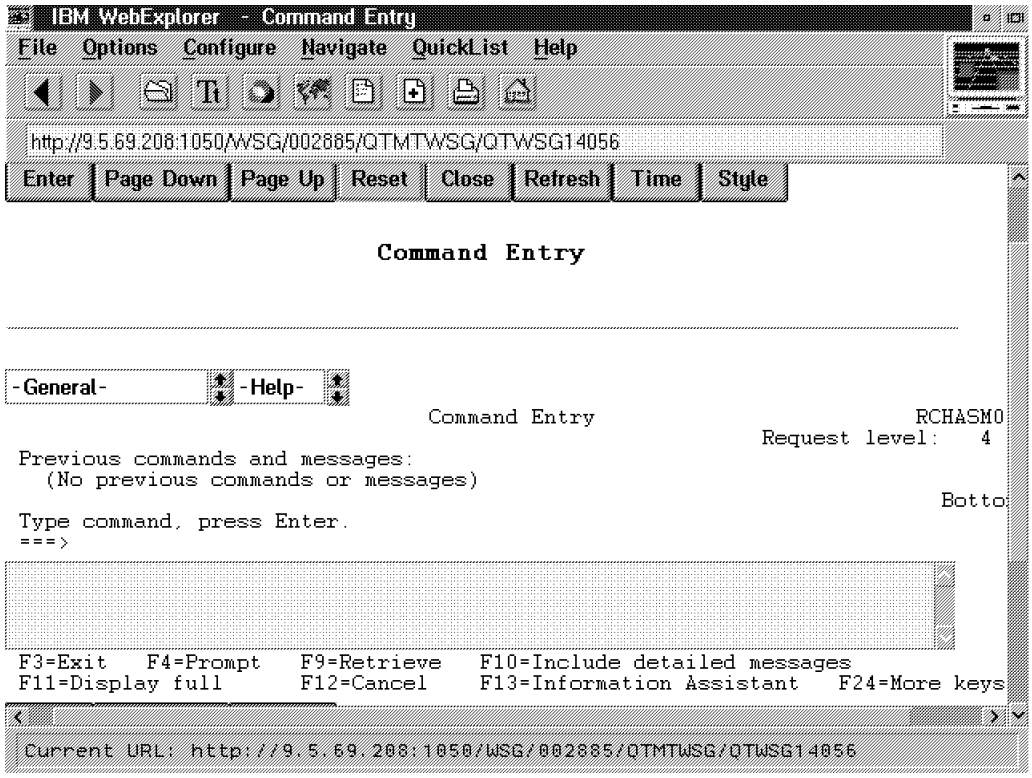


Figure 176. The Command Entry Display as seen by the Workstation Gateway

3. Work with Active Jobs:

   Figure 177 on page 238 shows the Work with Active Jobs display for the WebExplorer client.

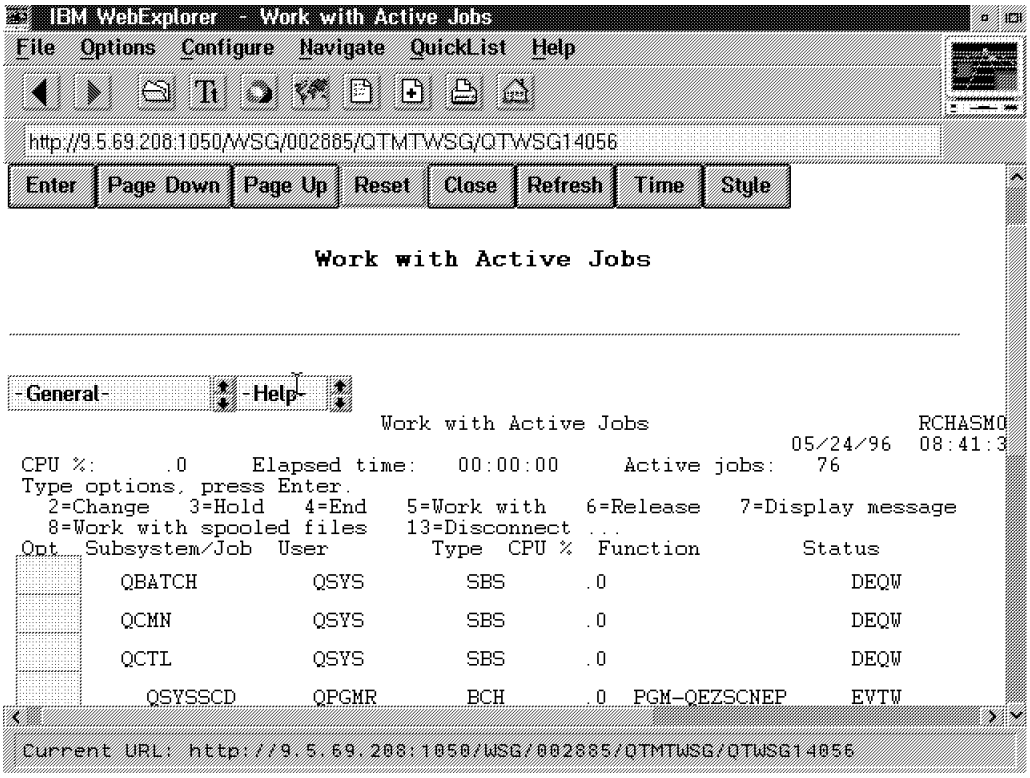


Figure 177. The Work with Active Jobs Display as seen by the Workstation Gateway

To summarize, using 5250 HTML gateway can give you the following benefits:

1. Web access to thousands of existing 'green-screen' applications.
2. Existing displays can be modified to include graphics, text, and links using the new DDS keyword "HTML" (displayed on Web client only).
3. Utilize existing tools and skills to develop new Web Applications.
4. Clients are no longer restricted to a particular emulator or operating system.
5. Complex web applications that need to save state through a series of steps can easily be done in DDS and 5250.

9.3.1 The 5250 HTML Gateway Server

Do not confuse the 5250 HTML Gateway with the HTTP Web Server. The HTTP Web Server allows the AS/400 system to act as a WWW server in the Internet. The 5250 HTML Gateway converts your 5250 data stream to HTML. Both can be started and function independently from each other.

The 5250 HTML Gateway is a TCP/IP application that services requests from HTTP clients. After the initial request is received from a client, that client is considered "active" and all future connections requests for that client occur over an arbitrary port number.

The client remains active until the session is signed off or an inactivity timeout limit is reached.

Note This

The 5250 HTML Gateway maintains the illusion that the browser is logically connected to the AS/400 system even though every transaction between the browser and the AS/400 server is disconnected. The AS/400 server maintains the virtual terminal API connection indefinitely or until the browser logs off or the inactivity timeout value is exceeded.

The 5250 HTML server is started through the STRTCPSVR SERVER(*WSG) command and ended with the ENDTCPSVR SERVER(*WSG) command.

Be sure to check the workstation gateway attribute to see if the Display Signon (DSPSGN) parameter is *YES or else you do not get the Signon display. You can use CHGWSGA command and then prompt to check it.

After you start or end the 5250 HTML server, you can use WRKACTJOB SBS(QSYSWRK) command to check the QSYSWRK subsystem to see whether the QTWSGnnnn jobs are there or not (′nnnnn′ is a unique numeric string that is derived from the timestamp).

Alternatively, it is started through the AUTOSTART option of the STRTCP command. The jobs are named ″QTWSGnnnnn″ where ″nnnnn″ is a unique numeric string that is derived from the timestamp.

The format of a link in an HTML document is called a Universal Resource Locator (URL). For HTTP, the URL identifies the protocol that the browser should use when contacting the server (for example, HTTP, FTP, WAIS, Gopher, and so on), the location of the server, and of the requested object. HTTP has the following form:

```
http://hostname:port/path
```

The port numbers for most TCP/IP applications such as FTP, Telnet, or WWW are predefined or you might say ″well-known″ numbers, which means everyone knows them and uses the same port numbers.

The 5250 HTML Gateway does not have such a well-known port number such as the HTTP server has. Therefore, the port number used by the AS/400 Workstation Gateway is found by querying the local TCP/IP configuration services database. To establish a 5250 HTML Gateway session, you must connect to it by using the form:

```
http://hostname:port/WSG
```

where port is the configured port number for that 5250 HTML Gateway. The default is a TCP port of 5061.

The 5250 server is organized into:

- A single *parent* job that *listens* and *accepts* connections from HTTP browser clients. It is important to note that the port used by 5250 HTML Gateway is different from the port of the HTTP Server because the 5250 HTML Server is a new type of server for which there is no well-known port. The parent job has only one function - to hand off connection requests to child jobs.
- One or more *child* jobs: A child job performs the actual work to satisfy the client connect request.

This technique allows you to do a multiplexing of connections within a single batch job.

Hints for using HTML gateway function:

1. The 5250 HTML gateway function is not a substitute for an original workstation connection.
2. The performance of using 5250 HTML gateway is not as good as using a traditional 5250 connection such as tn5250 through Telnet. Care should be given when sizing an AS/400 system as this kind of 5250 to HTML conversion is not done without extra cost in CPU terms.

   AS/400 response time should be expected to be .5 to 6.0 seconds longer (depending on AS/400 processor speed, system load, and application screen I/O implementation) than when using a normal 5250 interface. Response time components include the browser, communication line, and AS/400 CPU processing.

   Long response times are typically ″expected″ by users of the Internet. However, if the end user of an application has previously used normal 5250 connections and now uses the 5250 WSG support they could receive much longer response times. You should do a small benchmark to set proper performance expectations.

   If you are an IBMer...

   Performance tips for minimizing the 5250 WSG response time are documented in the *Performance Capabilities Reference for PowerPC Technology (V3R7)* which can be obtained by entering the following on a VM command line:

   ```
   TOOLCAT MKTTOOLS GET AS4PPCPF PACKAGE
   ```

   To obtain equivalent CISC system performance information enter the following on a VM command line:

   ```
   REQUEST V3R2 FROM FIELDSIT AT RCHVMW2 (yourname
   ```

3. With the exception of the character input fields, always use the mouse to click on the function key you want instead of using a key on the keyboard such as Enter or F3.
4. Context sensitive help versus general help:

   The AS/400 system considers the ′?′ character as the **first** character of an input field to be a ″cursor move″ request. The request moves the cursor to that input field, thus allowing you to use the F1 (contextual help) or F4 (prompt) buttons. In fact, any function key button can be used, not just F1 and F4 buttons.

   The AS/400 system does not keep the ′?′ in the input field upon return from the function key button action. Please note that this help function applies only to input fields. If you have help defined for some output fields, there is no way to get to it.

   For general help, the Help menu pull-down (or button) first moves the cursor to display row 1, column 1, and then invokes the help command. This is **not** the same as context sensitive help. This is because the cursor is moved to a position that is probably context insensitive before invoking help. This also means using the Help menu pull-down (or button) does not normally give the help for the home (default) input field, and the ′?′ invocation mechanism may be needed to force this help to appear.

5. Different web browsers may have a different display output when handling the same HTML tags.
6. After you have established a session between your web client and the AS/400 HTML gateway server function, you may start a Telnet or STRPASTHR session to another remote host. This function was not formally tested, but it seems to work in our ITSO network.
7. Never use previous and next page function in a web browser as a way to jump into the application flow from that page. We suggest turning off caching on the web client to enforce this suggestion.
8. No text assist.
9. No applications that automatically update the 5250 display without the client using an aid key. An example is the performance tools WRKSYSACT command when you use the automatic display refresh option.
10. Style button:

    The action bar (top row of buttons) has a style button that toggles the style used for the F1-F24 buttons. One mode shows displays F1-F24 as two rows of buttons at the bottom of every display. This takes up more display space, but has the advantage of letting you quickly submit the form.

    The other mode puts F1-F24 into a Function menu pull-down next to the General menu pull-down. This makes submitting the F1-F24 keys a two-step process (select menu item, then press Enter), but has the advantage of less display clutter and a faster display by the browser.
11. NLS code pages:

    Special characters and code pages can be requested by each individual client. Translation is usually from the EBCDIC CCSID of your AS/400 system to the ASCII CCSID specified in the CHGWSGA CCSID parameter. So, if the CHGWSGA CCSID parameter is changed, the EBCDIC CCSID default value also changes, since we get the ″best fit″ EBCDIC CCSID from the configured ASCII CCSID. This default can be overridden by each user. Examples for Sweden are:

    ```
    No user exit:
        http://as400.endicott.ibm.com:5061/WSG-SWB
        http://as400.endicott.ibm.com:5061/WSG-SWI
        http://as400.endicott.ibm.com:5061/WSG-SFI

    User exit:
        http://as400.endicott.ibm.com:5061/WSG-SWB/QAPP0100
        http://as400.endicott.ibm.com:5061/WSG-SWI/QAPP0100?any_string_data
        http://as400.endicott.ibm.com:5061/WSG-SFI/QAPP0100
    ```

    Refer to Appendix C in the National Language Support, SC41-3101, for a table of supported keyboard strings that can be used.

    Here are the rules to which the WSG will convert EBCDIC to ASCII:

    - The WSG builds the HTML in codepage 037 using only invariant characters for the tags and control words.
    - The 5250 application sends the data/character in the code page that it determine independent of the WSG.
    - Once the document is built, it is converted to ASCII in the CCSID specified in the following order:

      a. MIME header specified by the remote browser

      b. Overridden by the interactive subsystem (WSG-xxx).

c. WSG attribute (CHGWSGA command)

d. The system default CCSID

12. If the remote web client will be signing on to the AS/400 with a unique user profile (not a shared anonymous profile) you can still use the OUTQ parameter of their user profile to direct all the print outs to a single queue. This queue, in turn, can be a remote output queue that uses TCP/IP to route the print data back to any printer in the network. Most likely the printer sitting right next to the user's desk. Many IBM manuals and redbooks have instructions on how to configure such remote output queues. One in particular that includes sample configurations is *Printer Device Programming* (SC41-3713-01).
13. It is possible to off load some of the CPU cost needed to convert 5250 applications (data streams) to HTML by placing an intermediate AS/400 system between the web client and the target AS/400 system which is really running the 5250 application. That is, the web client will access an intermediate AS/400 system via HTTP/HTML through the 5250-WSG and then immediately Telnet (or more efficiently STRPASTHR) to the target AS/400 system where the interactive 5250 application will be run.

    This has some interesting side-effects:

    - The intermediate AS/400 system will have communication tasks, WSG server jobs, and user jobs. All these will run at non-interactive speeds with the exception of the user jobs that will still be interactive.
    - If these users are only doing WSG transactions and not sneaking in any other interactive transactions on the intermediate system, then their user jobs will sit nearly idle. If so, then nearly all of the CPU used on the intermediate system will be non-interactive for which the AS/400 Server models would be the best price/performance.
    - While the intermediate system will off load the 5250 to HTML translation from the target AS/400, additional response time delay of at least 0.2 seconds (or more) should be anticipated because of the intermediate system.

9.3.1.1 Configure TCP/IP Workstation Gateway (CFGTCPWSG) Main Menu

The easiest way to configure the 5250 HTML gateway is to use the menus. The following examples show the sequence of the configuration commands.

The following display appears if the CFGTCPWSG command is entered on the command line, or if CFGTCPAPP option 15 is selected.

```
                Configure TCP/IP Workstation Gateway
                                                          System:  SYSNM011
 Select one of the following:

     1. Change workstation gateway attributes

  Related options:
    10. Configure HTTP
    11. Work with autoconfigure virtual devices
    12. Work with limit security officer device access

Selection or command
===> ____________________________________________________________________

F3=Exit   F4=Prompt   F9=Retrieve   F12=Cancel
```

Figure 178. CFGTCPWSG Display

- Option 1 - Prompts the CHGWSGA CL command.
- Option 10- Calls the CFGTCPHTTP CL command.
- Option 11- Calls WRKSYSVAL SYSVAL(QAUTOVRT).
- Option 12- Calls WRKSYSVAL SYSVAL(QLMTSECOFR).

9.3.1.2 Change Workstation Gateway Attributes (CHGWSGA) Command Prompt

The following display appears if the CHGWSGA command is prompted from the command line or if CFGTCPWSG option 1 is selected.

The values shown are the current values as determined by the Prompt Override Program for CHGWSGA.

```
       Change Workstation Gateway Attributes (CHGWSGA)
                                                          System:  SYSNM011
Type choices, press Enter.

Autostart  . . . . . . . . . . .   *YES          *NO, *YES, *SAME
Number of clients per server job   20           1-50, *SAME, *DFT
Inactivity timeout . . . . . . .   10           0-60 minutes, *SAME, *DFT
Data request timeout . . . . . .   10           1-1200 seconds, *SAME, *DFT
Display sign on panel  . . . . .   *YES          *SAME,  *NO,  *YES
Access logging . . . . . . . . .   *YES          *SAME,  *NO,  *YES
Top banner URL . . . . . . . . .   *NONE_____________________________________
_____________________________________________________________________________
Bottom banner URL  . . . . . . . > 'http://internut.rchland.ibm.com/ITSOIC.400/
as400.gif'

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

Figure 179. Change Workstation Gateway Attributes Display

The Display sign on panel is important to security. The default is *NO, which will not allow the AS/400 signon panel to be sent to a remote web client.

Also related to security is the Access logging option. The default is *NO. If *YES is specified, access log records will be placed in physical file QATMTLOG in library QUSRSYS.

The Top and Bottom banner URL, if used, will cause an image to be placed on every 5250 screen including the sign on screen when seen by a graphical web browser. We suggest that you do not specify a Top banner URL to keep most of the important 5250 screen information near the top of the WSG screen. In addition, you will notice that the URL specified in Figure 179 on page 243 is fully qualified in that it starts by telling the web client to use the HTTP protocol to go back to your AS/400 server to retrieve as400.gif. The reason this must be fully qualified is that even though the WSG is using the HTTP protocol, it is *not* an HTTP server. Web clients, by default, will go back to the same server where they got the HTML - which in this case will the the WSG at port 5061 (or some other port defined for the WSG child job usually in the 1000s). In this case you will need to fully qualify the URL to force the web client to go back to the AS/400 HTTP server.

Since many clients can be expected to use the 5250 HTML Gateway Server, it is important to try to always have free servers waiting for new connect requests. To stay ahead of potential load demands, jobs are ″pre-started″ to avoid SBMJOB latency when a new server job is close to being needed.

When we say ″pre-started″, we mean that we submit a new child server with the SBMJOB when the number of available jobs (remember we are multiplexing connections within a single batch server job) goes below threshold limits. The threshold limit is determined based upon the value selected for the configured number of clients.

We have two types of timeouts for the 5250 HTML Gateway Server:

1. Inactivity timeout (INACTTIMO) - default 10 minutes:

   Specifies the number of minutes the system allows a Workstation Gateway session to remain inactive before it is ended. When a WSG session is inactive longer than the specified length of time, it is ended.

   Note: It may take the system an additional 1 to 120 seconds to end the inactive session.

   If WSG session is ended by the system, you see a message on the Web Browser display ″Session in use - perhaps your session expired?″ when you try to continue the operation.

2. Data request timeout (DTARQSTIMO) - default 10 seconds:

   Specifies the number of seconds from the time a browser connects until the client′s request data is received by the Workstation Gateway.

9.3.1.3 What Happens with My Existing Display Files?

Your existing display files need not be changed. You can use all DDS specifications as you did before. The DDS becomes (when compiled) a 5250 data stream. This means that the DDS keywords such as DSPATR(UL), BLINK, CHECK, and so on are translated in a coded string of data. In this data string, each field is preceded by one or more attribute bytes. This information makes a field such as a customer name underlined, protected, or blinking.

The AS/400 system (or more precise, the twinax workstation IOP (input/output processor)) sends out this generated 5250 data stream to your 5250 ″green-screen″. The hardware of your display then interprets this stream of data and produces a protected, underlined, or blinking field on your display.

This is how it works today. With V3R2, the 5250 HTML gateway intercepts this 5250 data stream and converts it ″on the fly″ to an HTML data stream. Let′s look at an example to make it more comprehensive.

First, we show you a simple DDS example of a display and how it looks on a 5250 workstation (″green-screen″).

Note: This DDS example is not using any new techniques or HTML keywords.

```
Columns . . . :    1  71           Edit                        ITSOIC400/DDSS
SEU==>                                                                WSGDSP
FMT DP.....AAN01N02N03T.Name++++++RLen++TDpBLinPosFunctions+++++++++++++++++
0012.00     A          R DSPR2
0013.00     A                                    2 34'ITSO Company'
0014.00     A                                    3 25'Customer Comment +
0015.00     A                                         Inquiry System'
0016.00     A
0017.00     A                                    6 24'Name : '
0018.00     A            FNAME_X       25   0  6 32
0019.00     A N02                             7 22'E-mail : '
0020.00     A N02        EMAIL_X       30   0  7 32
0021.00     A N02                             8 20'Address1 : '
0022.00     A N02        ADDRESS1_X    25   0  8 32
0023.00     A N02                             9 20'Address2 : '
0024.00     A N02        ADDRESS2_X    25   0  9 32
0025.00     A N02                            10 24'City : '
0026.00     A N02        CITY_X        25   0 10 32
0027.00     A N02                            11 23'State : '
0028.00     A N02        STATE_X        3   0 11 32

 F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor   F11=Toggle
 F16=Repeat find       F17=Repeat change          F24=More keys
```

Figure 180. DDS Source for Customer Comment Inquiry Display

The preceding DDS looks the same as this on a 5250 display station:

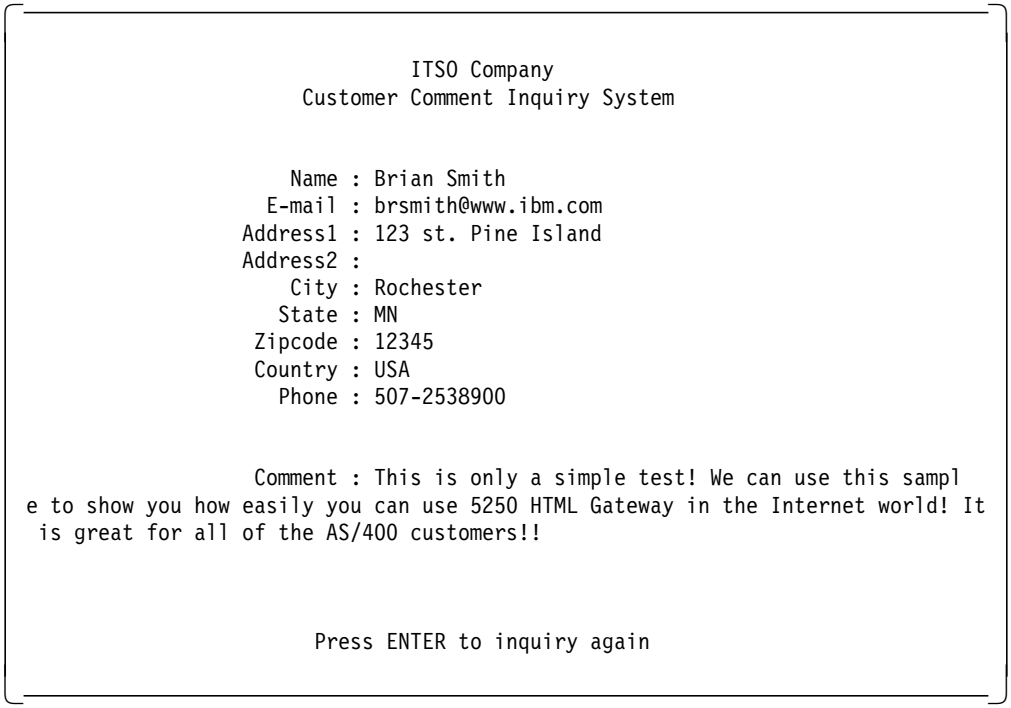


Figure 181. Customer Comment Inquiry DDS on the Traditional Text 5250 Display

Now let's see what the 5250 HTML gateway made out of our DDS specifications. The following display shows the result of the 5250 data stream conversion process. Note that this does not mean that you had to recompile the display file. The 5250 HTML gateway did this automatically "on the fly" for you. When the 5250 HTML gateway detected that the terminal that receives the 5250 data stream is a "virtual terminal" (that is, Web client), the 5250 data stream was converted to the HTML data stream.

```
<HTML>
<HEAD>
<TITLE>AS/400 Workstation Gateway </TITLE>
</HEAD>
<BODY>
<FORM METHOD="POST" ACTION="http://9.5.69.208:1029/WSG/003816/QTMTWSG/
<INPUT TYPE="HIDDEN" NAME="SESSION" VALUE="/A27CDCB57704125F/3B35D6E8">
VALUE="Enter"><INPUT TYPE="SUBMIT" NAME="/A27CDCB57704125F/3B35D6E8/
NAME="/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@u" VALUE="Page Up">
VALUE="Close"><INPUT TYPE="SUBMIT" NAME="SPECIALS" VALUE="Refresh">
VALUE="Style">
<CENTER><H3>AS/400 Workstation Gateway</H3></CENTER>
<HR>
<SELECT NAME="-General-" SIZE=1><OPTION SELECTED VALUE="-NONE-">-Gener
VALUE="@C">Clear<OPTION VALUE="@A@<">Record Back<OPTION VALUE="@x">PA1
VALUE="@A@C">Test Request<OPTION VALUE="@S@E">Host print screen<OPTION
VALUE="@A@H9O@E">Sign off<OPTION VALUE="@c">F12<OPTION VALUE="@3">F3</
VALUE="@H">Help</SELECT>
```

Figure 182 (Part 1 of 2). A Portion of the HTML Automatically Generated by the HTML Gateway

```
                              ITSO Company
                         Customer Comment Inquiry System
                            Name :  Brian Smith
                          E-mail :  brsmith@www.ibm.com
                        Address1 :  123 st. Pine Island
                        Address2 :
                            City :  Rochester
                           State :  MN
                         Zipcode :  12345
                         Country :  USA
                           Phone :  507-2538900
                         Comment :  This is only a simple test“ We can use
e to show you how easily you can use 5250 HTML Gateway in the Internet
 is great for all of the AS/400 customers”
                              Press ENTER to inquiry again
<INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@v” VALUE=”Page Down”>
Up”>
<INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@2” VALUE=”F2 ”><INPU
TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@4” VAL
VALUE=”F5 ”><INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUT
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@7” VALUE=”F7 ”><INPU
TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@9” VAL
VALUE=”F10”><INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUT
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@c” VALUE=”F12”>
<INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@e” VALUE=”F14”><INPU
TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@g” VAL
VALUE=”F17”><INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUT
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@j” VALUE=”F19”><INPU
TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@l” VAL
VALUE=”F22”><INPUT TYPE=”SUBMIT” NAME=”/A27CDCB57704125F/3B35D6E8/BUT
NAME=”/A27CDCB57704125F/3B35D6E8/BUTTON.999-999=@o” VALUE=”F24”>
</PRE>
</FORM>
</BODY>
</HTML>
```

Figure 182 (Part 2 of 2). A Portion of the HTML Automatically Generated by the HTML Gateway

Finally, let's see how this looks on an OS/2 web browser and also an example of how a subfile window looks.

Note: The result you see on a web browser is totally dependent upon how you configured the browser. If you choose another font, another background color, or another font size, the actual appearance of your HTML data stream on your PC might look quite different from our example.

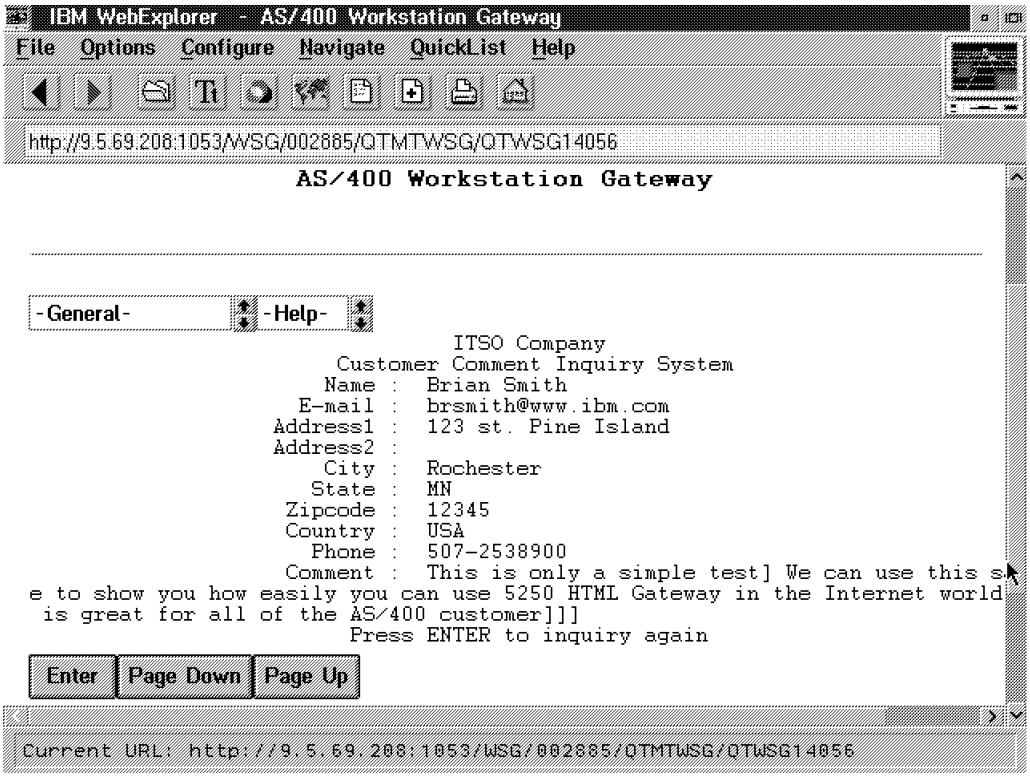


Figure 183. Display of Customer Master Record through 5250-HTML Gateway

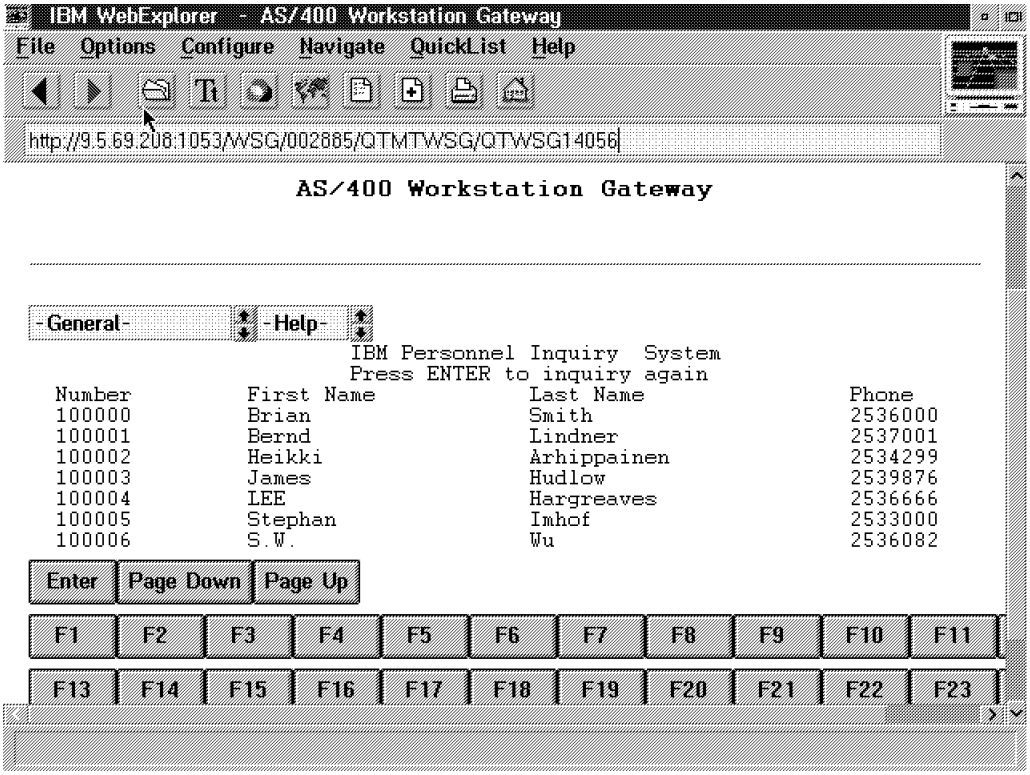


Figure 184. Display of Subfile Record through 5250-HTML Gateway

9.3.2 How To Enhance Your 5250 Applications with the DDS HTML Support

The 5250 HTML gateway support allows the insertion of HTML tags into the DDS of a display file. This allows us to utilize the graphic capabilities of a web browser with only minor changes to the existing DDS. For example, a customer can add graphics through the IMG HTML tag to an existing display file and display a graphic image along with the display.

Note: These HTML tags are only inserted into the data stream that flows to a terminal if the device query indicates that the device is a PC (or more precisely, an AS/400 5250 Workstation Gateway virtual terminal). Otherwise, the HTML tags are ignored for normal displays.

This simplifies and eases the handling of display files because only *one* source is needed for graphical workstations (that is, PCs) and ″green-screens″.

9.3.2.1 The New DDS Keyword

There is a new DDS keyword: HTML (HyperText Markup Language). This field-level keyword can be treated the same as a usual constant. Two things are different from a common constant. First, you have to put the new keyword HTML before the constant, and second, the ″constant″ itself must consist of an HTML string that must use the HTML syntax.

Let's take a look at a DDS example with HTML statements.

```
A                                   DSPSIZ(24 80 *DS3)
A                                   CA03(03)
A                                   CA12(12)
A     R DSPREC
A                                2 22'IBM Rochester Personnel Inquiry -
A                                     System'
A                                3  4HTML('<IMG SRC="//INTERNUT/+
A                                    /ITSOIC.400/AS400.GIF">')
A                                3  4HTML('<Table BORDER>')
A                                3  4HTML('<TR>')
A                                3  4HTML('<TH COLSPAN=2></TH>')
A                                3  4HTML('<TH>Table Demo</TH>')
A                                3  4HTML('</TR>')
A                                3  4HTML('<TR ALIGN=CENTER>')
A                                3  4HTML('<TH ROWSPAN=2></TH>')
A                                3  4HTML('<TH>First</TH>')
A                                3  4HTML('<TD>Row</TD>')
A                                3  4HTML('</TR>')
A                                3  4HTML('<TR ALIGN=CENTER>')
A                                3  4HTML('<TH>Second</TH>')
A                                3  4HTML('<TD>Row</TD>')
A                                3  4HTML('</TR>')
A                                3  4HTML('</TABLE>')
A                                3  4HTML('<IMG ALIGN=TOP +
A                                    SRC="//INTERNUT/+
A                                    /ITSOIC.400/RHAND.GIF">')
A                                3  4HTML('<A HREF="http://internut/+
A                                    class/wsg000.htm">Link-Here</A>')
A                                6 20'Please Input Employee Number : '
A       EMPNUM         6S 0B  6 52
A  02                                9 26'  First Name : '
A  02        FIRSTN        20   0  9 42
A  02                               10 26'   Last Name : '
A  02        LASTN         20   0 10 42
A  02                               11 26'Phone Number : '
A  02        PHONO          7S 00 11 42
A  02                               14 25'Not found - Please press ENTER'
A  02                                     DSPATR(RI)
A                                   21  5'F03=Exit'
A  02                               21 30'Press ENTER to inquiry again'
```

Figure 185. Sample 5250 DDS Enhanced with the HTML Tag

Note: The plain text is mixed with HTML tags.

Now, let's take a look at the display output of the sample DDS with the preceding HTML tags. You can see there is a "Link-Here" which corresponds to the HREF in DDS. Also, "Table Demo" is related to the TABLE tags.

You should be able to try this sample application yourself, as it was included in the CD-ROM as part of this book. From both a traditional 'green screen' 5250 emulator and any web client simply enter:

```
ADDLIBLE ITSOIC400
CALL SHTMLR
```

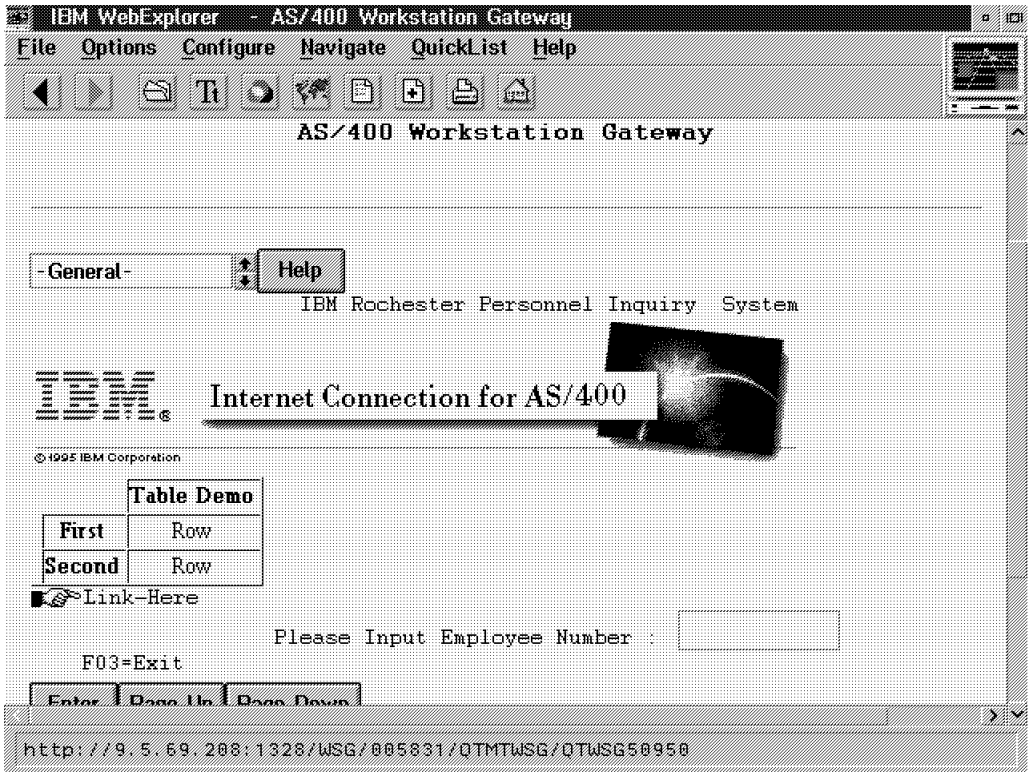


Figure 186. Link, Table, and Images imbedded in DDS

This display file and the program for it are in the library ITSOIC400. The name of the file is SHTMLD and its source is in the source file QDDSSRC. The name of the program is SHTMLR and its source is in QRPGSRC.

What are HTML Tags?

HTML documents consists of plain text interspersed with markup commands called **tags**. The tags are instructions to the browser software on how to display the text. They are represented by strings enclosed in <angle brackets> the same as the words before. See chapter 8.1, "HTTP Server" on page 159 for more information about HTML.

Another thing to mention is that in the preceding example, "normal" DDS keywords and HTML specs are used within one source.

HTML is a tag language where the order of the tags determines when they are processed. Row and column have no meaning in such a tag language. In this

case, the row and column are used to determine the order in which the HTML tags are sent to the browser.

With the HTML keyword, constant fields that have the same row and column value are processed in the order in which they appear in the DDS source.

Please notice that you can use any HTML tag except <HTML>, </HTML>, <HEAD>, </HEAD>, <TITLE>, </TITLE>, <BODY>, and </BODY> because all of the preceding tags are used by HTML Gateway to build the overall structure.

How to Determine if HTML is Processed?

On the CRTDSPF command, the ENHDSP (enhanced display) parameter is used to ignore or process the HTML keywords. This setting can be changed dynamically.

9.3.2.2 The New HTML Specification

The new HTML specification can have two formats:

```
HTML("datastring with a valid HTML tag")
```

or

```
HTML(&program-to-system-field)
```

A parameter is required after an HTML keyword. This parameter can be a valid HTML tag enclosed in single quotes or a program variable. The program-to-system field can be any legal length and has to be alphanumeric (A in position 35).

Note: The syntax of the HTML tag is *not* syntax checked by the DDS compiler. The browser that receives the HTML sequence performs syntax checking.

9.3.2.3 Limitations and Restrictions

The following keywords are not allowed with the HTML keyword:

- COLOR
- DATE
- DFT
- DSPATR
- EDTCDE
- EDTWRD
- HLPID
- MSGCON
- NOCCSID
- OVRATR
- PUTRETAIN
- SYSNAME
- TIME
- USER

The HTML keyword is not allowed on a field in a subfile record.

9.3.3 Additional Publications on the Web

URLs:

<**http://www.as400.ibm.com/ncc/webconn/htmlgate.htm**>
What is the WSG? This web page with the latest news about the WSG server and supported by the developers is available on the Internet.

Another avenue for retrieving late breaking news is Information APARs. At the time this document was printed, 5763-TC1 Information APAR II09450 contains the latest information.

<**http://www.ncsa.uiuc.edu/demoweb/url-primer.html**>
A Beginner's Guide to URLs

<**http://www.ics.uci.edu/pub/ietf/http/**>
IETF - Hypertext Transfer Protocol (HTTP) Working Group

<**http://info.cern.ch/hypertext/WWW/Technical.html**>
Technical Aspects of the World-Wide Web

<**http://www.w3.org/hypertext/WWW/Daemon/Overview.html**>
World-Wide Web server software

<**http://elanor.oit.unc.edu/http-prob.html**>
Analysis of HTTP Performance Problems

<**http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/mogul/HTTPLatency.html**
Improving HTTP Latency

<**http://www.ncsa.uiuc.edu/InformationServers/Performance/V1.4/report.html**
Performance of Several HTTP Demons on an HP 735 Workstation

http://www.w3.org/
The World Wide Web Initiative: The Project

9.3.4 5250 HTML Workstation Gateway Application Logon Exit Program

An application logon exit program (QAPP0100) allows bypassing the AS/400 Signon display and invoking an application program directly without the client browser having to send a user profile or password. This allows the customer the option of providing *any* application to client browsers without requiring a Signon. This has the added benefit of removing the command line for this client. This is done by calling a customer program that authenticates the client request and provides Signon information to the 5250 HTML Gateway Server.

The 5250 HTML Gateway Server uses the output of the customer's User Exit and performs the Signon action on behalf of the client browser.

When the user exit is given control, it may perform any desired validation using the supplied Internet Protocol address and any of the supplied operation-specific information extracted after the ″/WSG/QAPP0100″ string in the URL. Setting the ″Allow Operation″ output determines whether the automatic logon is performed, or whether an error message is returned to the client browser.

If the operation is allowed, then the user exit must return the user profile, password, current library, and program. All output must be non-NULL or else an error is returned to the client browser.

Just use the following steps to use the Exit Program:

1. Code your Exit Program. You can use the sample program provided in library ITSOIC400, file QCSRC, member EXITPGM2.
2. End the 5250 HTML server by using the ENDTCPSVR SERVER(*WSG) command.

3. Register your Exit Program by using the WRKREGINF EXITPNT(QIBM_QTMT_WSG) command. Then use option 8 to add your exit program. If you have used the install program that came with the CD-ROM, this might have already been done for you.
4. Start the 5250 HTML server by using the STRTCPSVR SERVER(*WSG) command.
5. Open a URL in web browser as http://hostname:5061/WSG/QAPP0100?anon where anon is the string that you want a user to type in to allow them direct access to the SHTMLR application.

9.3.4.1 Sample Exit Program for HTML Gateway

This sample exit program uses one of the input variables (IP address of the remote host system) from the HTML Gateway system program to check if a web browser user has the right to use the HTML Gateway function.

You can find the source of this sample program in the QILECSRC source file in the ITSOIC400 library.

From Web Browser, open the URL http://hostname.domainname:5061/WSG/QAPP0100. Then you can bypass the signon display and go into the command entry display.

```
/* Module Description ************************************************/
/*                                                                   */
/*                                                                   */
/* Source File Name: exitpgm2.c                                      */
/*                                                                   */
/* Module Name: Workstation Gateway Server logon exit program.       */
/*                                                                   */
/* Service Program Name:  n/a                                        */
/*                                                                   */
/* Source File Description:                                          */
/*                                                                   */
/*   This module contains functions to allow a client browser to     */
/*   bypass an AS/400 sign-on panel and invoke an application.       */
/*                                                                   */
/* Reference:                                                        */
/*   TCP/IP Configuration and Reference SC41-3420 Appendix-I.        */
/*                                                                   */
/* End Module Description ********************************************/


#define _EXITPGM_C
```

Figure 187 (Part 1 of 8). Sample Exit Program for HTML Gateway

```
/*********************************************************************/
/* All file scoped includes go here                                  */
/*********************************************************************/


#ifndef __stdio_h
#include <stdio.h>
#endif

#ifndef __string_h
#include <string.h>
#endif

#ifndef __stdlib_h
#include <stdlib.h>
#endif


/*********************************************************************/
/* All file scoped Constants go here                                 */
/*********************************************************************/


#define  SIZE    10
#define  FNAME   21              /* Qualified database file name size */
#define  FWIDTH  240             /* Width of one database file record */
#define  BLANK   ' '


/*********************************************************************/
/* All file scoped type declarations go here                         */
/*********************************************************************/
/* Structure for data passed to Server Logon exit program.           */
/*********************************************************************/


typedef struct
{
  char *OperSpecInfo_p;         /* Operation Specific Info    (Input) */
  int  Lgth_OperSpecInfo;       /* Operation Spec Info length (Input) */
  char ClientIPaddr[15];        /* Client IP Addr.            (Input) */
  int  CCSID;                   /* CCSID of operation info    (Input) */
  char AllowOper[1];            /* Allow Operation '0'=N,'1'=Y(Output)*/
  char UserProfile[SIZE];       /* User Profile.             (Output) */
  char Password[SIZE];          /* Password.                 (Output) */
  char ProgramLib[SIZE];        /* Library of program to run.(Output) */
  char ProgramName[SIZE];       /* Program to invoke.        (Output) */
  char InitialMenu[SIZE];       /* Initial menu to invoke.   (Output) */
} QAPP0100_I_t;
```

Figure 187 (Part 2 of 8). Sample Exit Program for HTML Gateway

```
/*********************************************************************/
/* All file scoped Macro invocations go here                         */
/*********************************************************************/

/*********************************************************************/
/*                                                                   */
/* Macro name: TRACE                                                 */
/*                                                                   */
/* Log test result entry to an output file for workstation gateway.  */
/*                                                                   */
/*   Arguments:            x - Text string of application-specific   */
/*                             trace data from the calling program.  */
/*                         y - Test results output file pointer.     */
/*                                                                   */
/*********************************************************************/


#define TRACE(x, y)                                                   \
 {                                                                    \
 memset(file_buff, BLANK, sizeof(file_buff));                         \
 sprintf(file_buff, "%s%c", x, '\0');                                 \
 fwrite(file_buff, FWIDTH, 1, y);                                     \
 }


/*********************************************************************/
/* All internal function prototypes go here                          */
/*********************************************************************/


static void  exitpgm
  (char *,int,char *,int,char *,char *,char *,char *,char *,char *);


static void WriteParms
  (char *,int,char *,int,char *,char *,char *,char *,char *,char *);


/*********************************************************************/
/* All file scoped variable declarations go here                     */
/*********************************************************************/


char file_name[FNAME];        /* Output results database file name    */
FILE *test_file;              /* Output results database file pointer */
char file_buff[FWIDTH];       /* Output results file buffer           */
```

Figure 187 (Part 3 of 8). Sample Exit Program for HTML Gateway

```
/* Function Specification ********************************************/
/*                                                                   */
/* Function Name: Main                                               */
/*                                                                   */
/* Descriptive Name: Application Logon exit program sample program.  */
/*                                                                   */
/*   This test exit program provides control over signon panels via  */
/*   the WSG server in the V3R2 release.                             */
/*                                                                   */
/* Notes: For V3R2 the "argv[]" parameters are "char *" by definition.*/
/*         Reference integers as "*(int(argv[1]))", for example.     */
/*         Consider the method for passing them back to the caller.  */
/*                                                                   */
/*   Dependencies:                                                   */
/*      WSG Applicaton Logon exit point QIBM_QTMT_WSG format QAPPO100 */
/*       was registered during WSG V3R2 installation.                */
/*                                                                   */
/*   Restrictions:                                                   */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/*   Messages:                                                       */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/*   Side Effects:                                                   */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/*   Functions/Macros called:                                        */
/*                                                                   */
/*       TRACE - Write one data record to test results file.         */
/*                                                                   */
/* Input:                                                            */
/*   chat * argv[1]        - Operation specific information          */
/*   int    argv[2]        - Length of operation specific information */
/*   char * argv[3]        - IP address of the remote host system.   */
/*   int    argv[4]        - CCSID of the operation specific info    */
/*   char * argv[5]        - Allow operation '0'=No, '1'=Yes(output) */
/*   char * argv[6]        - User profile to be used (output)        */
/*   char * argv[7]        - Password to be used (output)            */
/*   char * argv[8]        - Program library to be used (output)     */
/*   char * argv[9]        - Program name to be used (output)        */
/*   char * argv[10]       - Menu panel to be used (output)          */
/*                                                                   */
/* Exit Normal: Return AllowOper value to server application.        */
/*                                                                   */
/* Exit Error: None                                                  */
/*                                                                   */
/* End Function Specification ****************************************/


void main(int argc, char *argv[])
{
     exitpgm(argv[1],
             *((int *)(argv[2])),
             argv[3],
             *((int *)(argv[4])),
             argv[5],
             argv[6],
             argv[7],
             argv[8],
             argv[9],
             argv[10]);
  return;
}            /* End main */
```

Figure 187 (Part 4 of 8). Sample Exit Program for HTML Gateway

```
/* Function Specification ********************************************/
/*                                                                   */
/* Function Name: exitpgm                                            */
/*                                                                   */
/* Descriptive Name: Workstation Gateway Server (WSG) Logon exit.    */
/*                                                                   */
/*   This test exit program provides control over user authentication */
/*   to a workstation gateway in the V3R2 release.                   */
/*                                                                   */
/* Notes:                                                            */
/*                                                                   */
/*   Dependencies:                                                   */
/*                                                                   */
/*       Workstation Gateway Logon exit point QIBM_QTMT_WSG was      */
/*        registered during WEB V3R2 installation.                   */
/*                                                                   */
/*   Restrictions:                                                   */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/*   Messages:                                                       */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/*   Side Effects:                                                   */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/*   Functions/Macros called:                                        */
/*                                                                   */
/*       None                                                        */
/*                                                                   */
/* Input:                                                            */
/*  char * OperSpecInfo_p     - Operation Specific Information.      */
/*  int    Lgth_OperSpecInfo  - Length (in bytes) of Operation       */
/*                              Specific Information.                */
/*  char   ClientIPaddr       - Client Internet Protocol Address.    */
/*  int    CCSID              - CCSID of operation info              */
/*                                                                   */
/* Output:                                                           */
/*  char * AllowOper          - Allow Operation ('0' = Reject),      */
/*                                              ('1' = Accept).      */
/*  char * UserProfile        - User Profile to be used for sign on. */
/*  char * Password           - Password to be used for sign on.     */
/*  char * ProgramLib         - Library of program to invoke.        */
/*  char * ProgramName        - Name of program to invoke.           */
/*  char * InitialMenu        - Initial menu to invoke.              */
/*                                                                   */
/* Exit Normal: (See OUTPUT)                                         */
/*                                                                   */
/* Exit Error:  None                                                 */
/*                                                                   */
/* End Function Specification ****************************************/


static void exitpgm(char *OperSpecInfo_p,         /* Entry point */
                    int Lgth_OperSpecInfo,
                    char ClientIPaddr[15],
                    int  CCSID,
                    char AllowOper[1],
                    char UserProfile[SIZE],
                    char Password[SIZE],
                    char ProgramLib[SIZE],
                    char ProgramName[SIZE],
                    char InitialMenu[SIZE])

{       /* exitpgm start from here */
```

Figure 187 (Part 5 of 8). Sample Exit Program for HTML Gateway

```
/*********************************************************************/
/*                                                                   */
/* You can design your own logical flow here to check if user is     */
/* authorized to bypass sign on screen. Following are some examples.*/
/*                                                                   */
/* (1) Validate Client IP address.                                   */
/* (2) Parse the "OperSpecInfo_p" input string.                      */
/*                                                                   */
/* Then                                                              */
/* Return AllowOper of '0' - Reject this clients request.            */
/* Return AllowOper of '1' - Accept this clients request.            */
/* Set related return values for client browser.                     */
/*                                                                   */
/*********************************************************************/
/*                                                                   */
/******  Be sure to modify below for your environment  **************/
/*                                                                   */
/* Following is an example by checking the parameter coming from     *
/*                        the client                                 *
/* Accept the following string = "anon" (for anonymous)              *
/*                                                                   *
/*********************************************************************


 /* change here for your logic */

 char Accept_IP[] = "anon";

 if (strstr(OperSpecInfo_p, Accept_IP) && (Lgth_OperSpecInfo >= 4)){
     memcpy(UserProfile, "ANONYMOUS ", SIZE);
     memcpy(Password,    "ANONYMOUS ", SIZE);
     memcpy(ProgramLib,  "ITSOIC400 ", SIZE);
     memcpy(ProgramName, "SHTMLR    ", SIZE);
     memcpy(InitialMenu, "          ", SIZE);
     memcpy(AllowOper,   "1",          1);
  }  else {
         memcpy(AllowOper,   "0",          1);
  }


/*********************************************************************/
/* Call function to write input parameters as received to file.      */
/*********************************************************************/


 WriteParms(OperSpecInfo_p,
            Lgth_OperSpecInfo,
            ClientIPaddr,
            CCSID,
            AllowOper,
            UserProfile,
            Password,
            ProgramLib,
            ProgramName,
            InitialMenu);

 return;                  /* End program exitpgm.c               */

}    /* end of exitpgm */
```

Figure 187 (Part 6 of 8). Sample Exit Program for HTML Gateway

```
/* Function Specification ********************************************/
/*                                                                   */
/* Function Name: WriteParms                                         */
/*                                                                   */
/* Descriptive Name: Write the input parameters as received to file.  */
/*                                                                   */
/*                                                                   */
/* Notes:                                                            */
/*                                                                   */
/*   Dependencies:                                                   */
/*      None                                                         */
/*                                                                   */
/*   Restrictions:                                                   */
/*       None                                                        */
/*                                                                   */
/*   Messages:                                                       */
/*       None                                                        */
/*                                                                   */
/*   Side Effects:                                                   */
/*       None                                                        */
/*                                                                   */
/*   Functions/Macros called:                                        */
/*                                                                   */
/*       system                                                      */
/*       sprintf                                                     */
/*       TRACE - Write one data record to test results file.         */
/*                                                                   */
/* Input:                                                            */
/*  char * OperSpecInfo_p     - Operation Specific Information.      */
/*  int    Lgth_OperSpecInfo  - Length (in bytes) of Operation       */
/*                              Specific Information.                */
/*  char * ClientIPaddr       - Client Internet Protocol Address.    */
/*  int    CCSID              - CCSID of operation information       */
/*  char * AllowOper          - Allow Operation ('0' = Reject),      */
/*                                              ('1' = Accept).      */
/*  char * UserProfile        - User Profile to be used for sign on. */
/*  char * Password           - Password to be used for sign on.     */
/*  char * ProgramLib         - Library of program to invoke.        */
/*  char * ProgramName        - Name of program to invoke.           */
/*  char * InitialMenu        - Initial menu to invoke.              */
/*                                                                   */
/* Output:                                                           */
/*   A single, serial data record written to the test results file.  */
/*                                                                   */
/* Exit Normal: (See OUTPUT)                                         */
/*                                                                   */
/* Exit Error:  None.                                                */
/*                                                                   */
/* End Function Specification ****************************************/


static void WriteParms(char *OperSpecInfo_p,         /* Entry point */
                       int Lgth_OperSpecInfo,
                       char ClientIPaddr[15],
                       int  CCSID,
                       char AllowOper[1],
                       char UserProfile[SIZE],
                       char Password[SIZE],
                       char ProgramLib[SIZE],
                       char ProgramName[SIZE],
                       char InitialMenu[SIZE])

{
```

Figure 187 (Part 7 of 8). Sample Exit Program for HTML Gateway

```
  /*********************************************************************/
  /* Local Variables                                                   */
  /*********************************************************************/


  char loc_buff[FWIDTH];                        /* Local TRACE buffer */


  /*********************************************************************/
  /* Create a database file to write the logon attempt results to.     */
  /*********************************************************************/


  system("QSYS/CRTPF FILE(QUSRSYS/EXITPGM) RCDLEN(240) SIZE(*NOMAX)");


  /*********************************************************************/
  /* Build up the qualified database file name.                        */
  /*********************************************************************/


  sprintf(file_name, "%s", "QUSRSYS/EXITPGM");


  /*********************************************************************/
  /* Open database file for permanent log of test results.             */
  /*  mode = a(ppend) b(inary) 128-byte fixed-length records           */
  /*********************************************************************/


  test_file = fopen(file_name,"ab, lrecl=240, type=record, recfm=f");


  /*********************************************************************/
  /* Build up the input record for the test results database file      */
  /* Write parameters that were passed in to the output file:          */
  /*********************************************************************/


  sprintf(loc_buff,
          "IP: %s CCSID: %d Allow: '%c'  Profile: %s Password: %s "
          "Library: %s Program: %s Menu: %s  OperLen: %d Oper: >%s<",
          ClientIPaddr,
          CCSID,
          *AllowOper,
          UserProfile,
          Password,
          ProgramLib,
          ProgramName,
          InitialMenu,
          Lgth_OperSpecInfo,
          OperSpecInfo_p);
  TRACE(loc_buff, test_file);
  fclose(test_file);
  return;
}


#undef _EXITPGM_C
```

Figure 187 (Part 8 of 8). Sample Exit Program for HTML Gateway

9.4 Server Side Image map Support

Important note

If you would like to follow along with the examples in this and other chapters you first will want to install the ITSO Company demonstration and other web based applications from the CD-ROM that came with this redbook. Please see Appendix A, “Installing the ITSO Company Demo” on page 285 for instructions on how to get your AS/400 up and running right away.

Suppose you want people to be able to choose a car, check if a certain model is available, and make reservations from your Web server. What would be better than to show them a map of the country and allow them to click on any location and make the desired reservation? That is what a clickable image map can do. It can have many other uses also. For example, it helps users better understand a picture or diagram by allowing them to click on a component represented in the graphic to see an explanation. It also lets users move down to finer levels of detail on a map of a campus, building, or whatever you want.

Clickable image maps obviously work only with graphically-oriented Web browsers. You should always try to provide alternate text-based methods of accessing the same information.

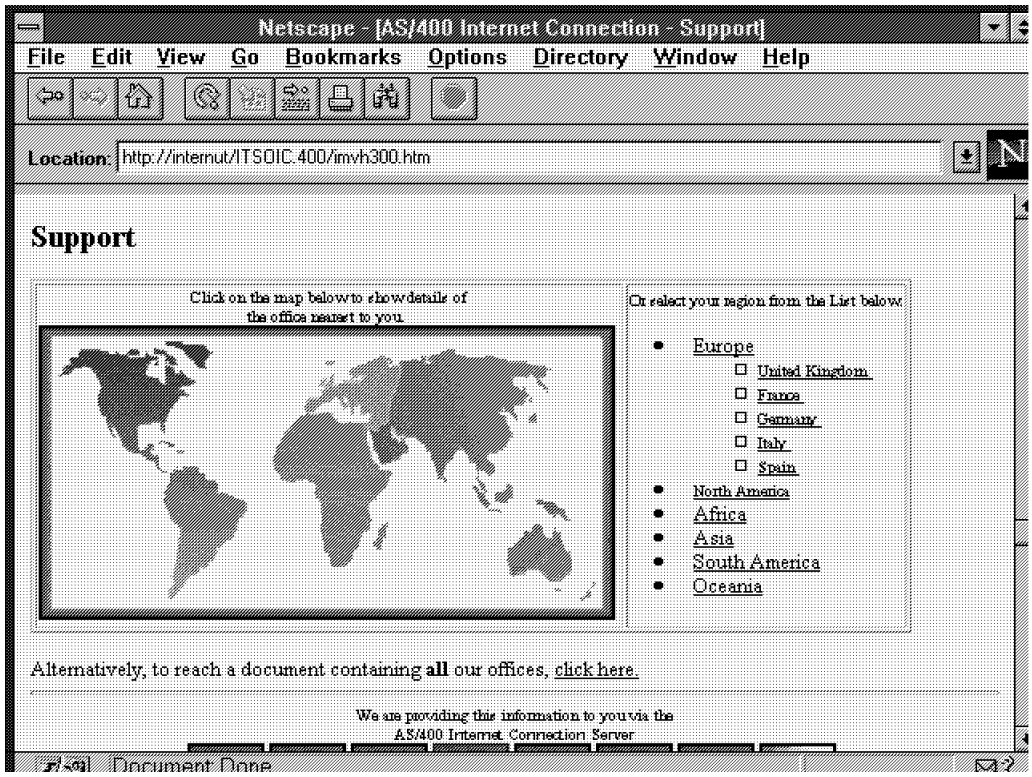


Figure 188. Clickable World Map. This figure shows a clickable map and an alternate text-based method of accessing the same information.

Image map requests are another form of program call or CGI. You can request image maps from either a server-provided image mapping program or a user-provided image mapping program. The easiest way to provide this level of function for your web application is to take advantage of a server-provided mapping program such as the one found on the AS/400 system. This IBM written CGI-BIN application is called QTMHIMAG in the QTCP library.

If an image map request is made from a user-provided image program, it is handled as a standard program call request that must meet the CGI interface. We do not show an example of the user-provided image mapping, as this is more work than it is worth.

There are several steps you need to take when creating a server-provided image map:

- Plan a clickable image map.
- Map the hotspots in a map file.
- Reference your clickable image map in HTML.
- Configure the HTTP server.

Let's take a look at each of these steps in detail.

9.4.1.1 Planning the Clickable Image Map

First, determine the image you are going to use. As for any inline images, the graphics need to be in GIF format for widest portability.

Then define where you are going to map the clickable areas or hotspots.

Depending on the size and make up of the image you are mapping, you can use polygons, rectangles, or circles to identify hotspots.

9.4.1.2 Mapping the Hotspots in a Map File

The next step is to develop an image map file such as the one seen previously. You need to create a separate image map file for each clickable image map on your Web server. Each line in an image map file represents a hot spot by defining an area within the graphic and the corresponding URL to be returned if someone clicks on that area with their Web browser. By using a program on your PC such as MAPEDIT, hotspots on the image are formatted in CERN by: Shape, Coordinate-1, Coordinate-2 ... Coordinate-n, URL.

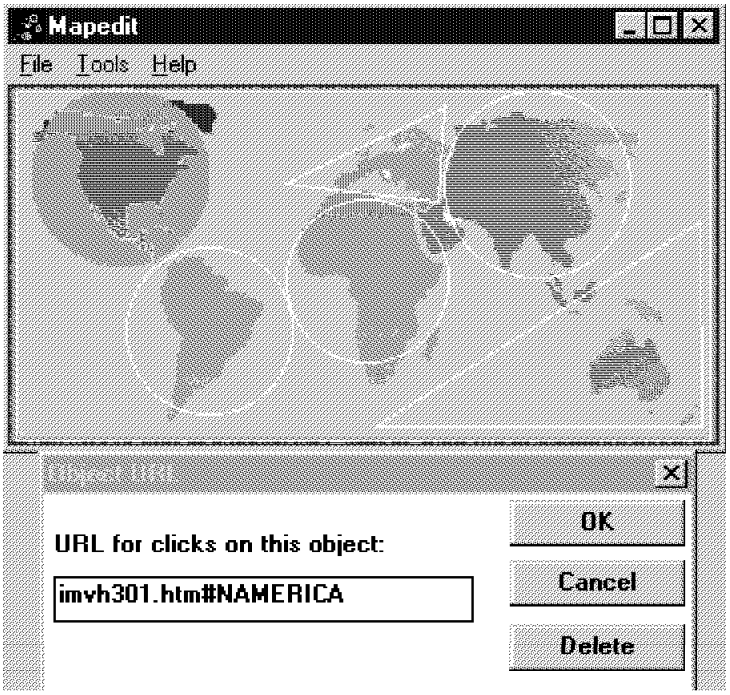


Figure 189. Defining Hotspots and Associated URLs. This figure shows North America in a circle shape mapped to its associated URL.

This map is made up of circles and triangular shaped polygons. A rectangular shape was mapped last for the entire image as a default area.

Shapes are defined by the following:

- A circle - for a circle (a pair of coordinates for the center and a single value for the radius).
- A polygon - for a polygon (the number of coordinates depend on the number of sides defined for the polygon).
- A rectangle - for a rectangle (with two coordinate pairs: upper-left and lower-right).

Coordinates are x,y pairs counting in pixels from the upper left-hand corner of the image. The URL can be either an actual URL pointing to any resource on the World Wide Web, or it can point to another image or document on your server.

```
      *************** Beginning of data ********************************
001.00 circle (101,123) 42 /ITSOIC.400/imvh301.htm#SAMERICA
002.00 circle (56,46) 45 /ITSOIC.400/imvh301.htm#NAMERICA
003.00 circle (180,98) 41 /ITSOIC.400/imvh301.htm#AFRICA
004.00 circle (265,50) 47 /ITSOIC.400/imvh301.htm#ASIA
005.00 poly (218,10) (215,58) (138,49) /ITSOIC.400/imvh301.htm#EUROPE
006.00 poly (346,68) (184,170) (347,170) /ITSOIC.400/imvh301.htm#OCEANIA
007.00 rect (4,3) (351,177) /ITSOIC.400/imvh301.htm#OCEANIA
008.00 default /ITSOIC.400/imvh301.htm#OCEANIA
      ****************** End of data ***********************************
```

Figure 190. Image Map Coordinates. This example shows circular, polygon, and rectangular shapes and their defined URLs.

The server takes the coordinates from a user's mouse click and steps through the map file to determine if the click is within any hotspots. As soon as the first

match is found, the corresponding URL is used to redirect a document to the user's Web browser. If no matches are found, a default URL (that you specify in the image map file) is returned.

Notice that the last line in the file is a default statement. An image map file must *always* contain a default statement. This is required so that if a user clicks on a part of the image that has not been mapped, the user is sent to a defined default area.

It is normal to mix and match circles, polygons, and rectangles in the same map file. Although you should try to minimize overlapping hotspots in the image map, if there are any, the first match is the one used.

9.4.1.3 Referencing Your Clickable Image Map in HTML

The final step in creating an image map is to tell the Web browser accessing the HTML document that the inline image it displays is a clickable image map. In practice, what you do is create a link where this image is the link trigger and the URL invokes the image map script and passes it the map name. For example:

```
<A HREF="http://servername/imagemap.pgm/imagemap.file">
<IMG SRC="image.gif"> ISMAP></A>
```

9.4.1.4 Configuring Your Clickable Image Map in the HTTP Server

In order to configure the HTTP server on the AS/400 system, we reference the previous example. Our `imagemap.pgm` is `qtcp/qtmhimag` and our image maps are found in the `imagemap.file`. Here is how our URL looks using the AS/400 system to serve image maps:

```
<A HREF="http://servername/qsys.lib/qtcp.lib/qtmhimag.pgm 1
/qsys.lib/itsoic400.lib/imagemap.file/mapk.mbr"> 2
<IMG SRC="imvg300.gif" ISMAP></A> 3
```

1 This section of the URL represents the server name and also where the image mapping program resides on the AS/400 system.

2 This section tells the server where the actual image map file is located on the AS/400 system.

3 This section shows the source of the image or GIF file that is to be mapped and ISMAP tells the server this is an image map.

We can shorten and simplify the URL by taking advantage of the HTTP mapping rules. For example, we use the following URL to do the same thing:

```
<A HREF="/cgi-bin/imagemap/map1.mbr">
<IMG SRC="world.gif" ISMAP></A>
```

Where you see `/cgi-bin/imagemap`, we used the HTTP `Map` configuration statement to reference the longer URL. The following lines must be added to the HTTP configuration to make it work:

```
Map /cgi-bin/imagemap/*  /qsys.lib/qtcp.lib/qtmhimag.pgm
/qsys.lib/itsoic400.lib/imagemap.file/*

Pass /qsys.lib/itsoic400.lib/imagemap.file/*

Exec /qsys.lib/qtcp.lib/*
```

In order to access the HTTP configuration file, type WRKHTTPCFG on an AS/400 command line. Add the `Map`, `Pass` , and `Exec` statements noted previously. By using the `Map` directive in our HTTP configuration, we are able to use shorter

URLs in our HTML documents. The Pass statement is there in order to pass the image map files from the server. There must also be an Exec statement in order to call the qtmhimag program from the qtcp library.

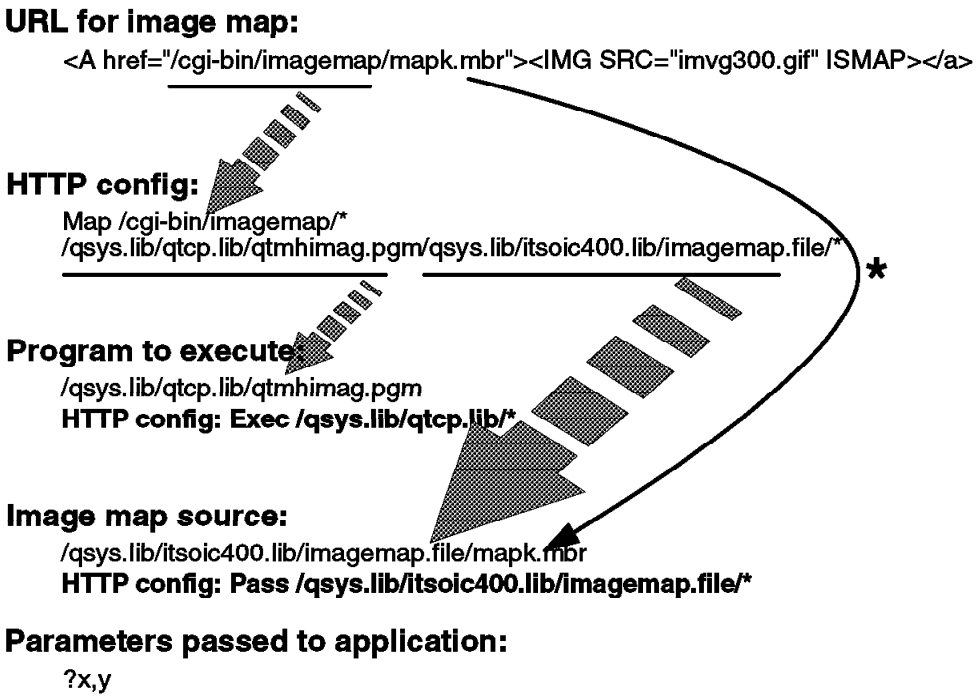


Figure 191. Mapping of Image Map URL to HTTP Configuration

Note

More information on image mapping and image map requests on the AS/400 system is available in the *TCP/IP Configuration and Reference, Version 3*, SC41-3420.

Chapter 10. Security and Audit on the Internet

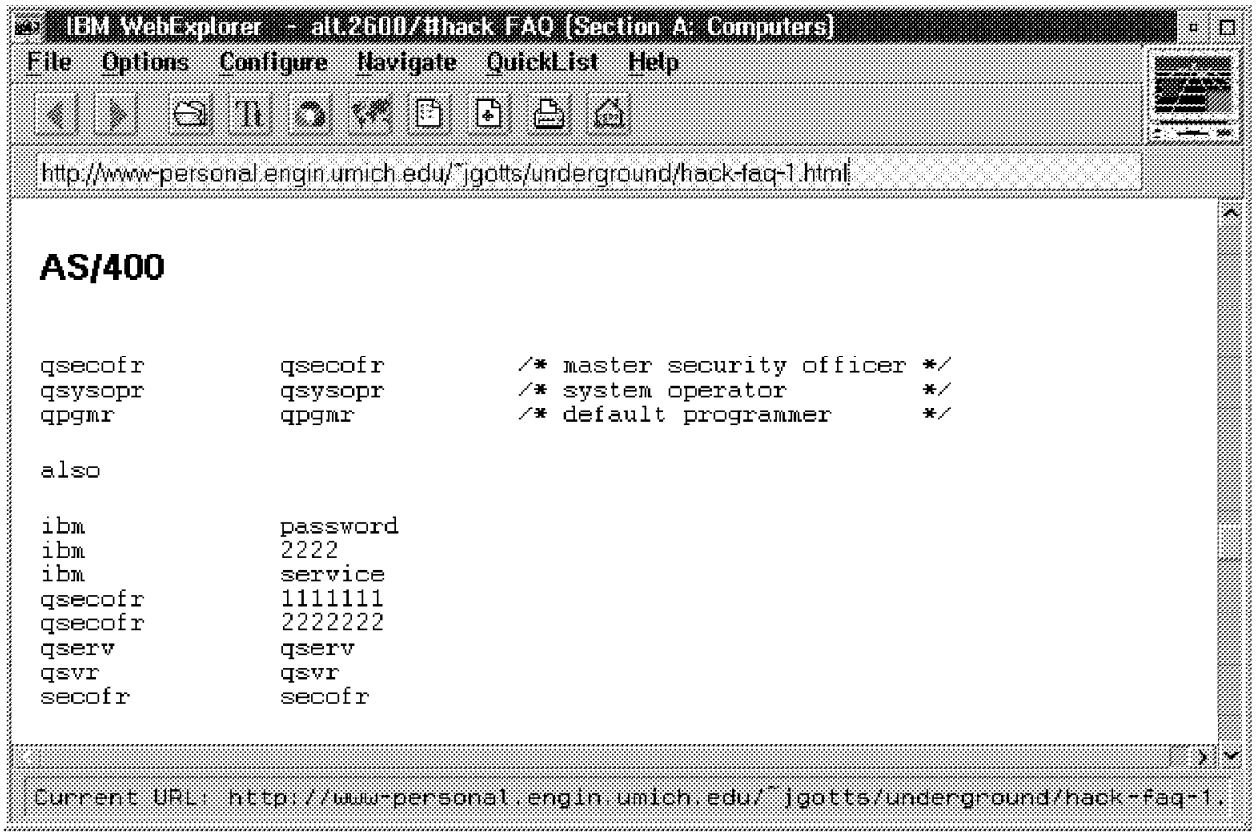


Figure 192. AS/400 Default Passwords Found in alt.2600/#hack FAQ!

Look out!

Figure 192 shows just how serious one should take Internet security. This home page shows common user ID's and passwords for AS/400 systems and other IBM and OEM system USERIDs and passwords. Many hackers attack a system using a dictionary of passwords, this clearly shows the AS/400 system is thought of as a target. The 1990 printing of Webster's Dictionary defines a hacker as "an expert at programming and solving problems with a computer". Anyone who is associated with computer security these days knows that hackers do not always apply their expertise in appropriate ways. This chapter addresses security concerns on the Internet and in particular, the AS/400 system.

Take a moment right now to order these two excellent books!

Tips and Tools for Securing Your AS/400 **(GC41-0615)** This book is a well written guide to the entire AS/400's security model. It also addresses how to secure specific TCP/IP applications on your AS/400.

IBM SecureWay AS/400 and the Internet **(G325-6321)** This book focuses on the attachment of your AS/400 to the Internet. It describes a series of network configuration scenarios and identifies the areas that you must address to provide a secured server on the Internet. Again, well written.

10.1 Some Background to Security

In the past, computer and network security used to be a simple thing. Companies maintained their own networks separate from the outside world. With this environment, it was easier to control who had and who did not have access to your corporate information. One of the biggest challenges of distributed computing is implementing effective security throughout the network. Security access when connected to a network can be summed up simply by considering these points:

- No Access -- Computer is a stand-alone system and not connected to any network with internal passwords and security.
- Firewall -- Computer is accessible from a network that is protected by internal passwords and security, and further by a Firewall separating the network from the Internet.
- All Access -- Computer is accessible from a network that is protected by internal passwords and security but it is directly connected to the Internet.

There is always a trade-off to be made between making a computer secure and the function it can provide. In the extreme case, the most secure is when the computer is turned off. By connecting to the Internet, you can open the door to potential cases of lost data, programs, or jeopardizing the privacy of your business and your employees. When computers are networked, the problems are compounded since the communication channel itself is open to attack. These attacks can be summarized in two ways:

1. Passive attacks: tapping or tracing your communications. These are very difficult to detect. You should always assume someone is looking at every transaction you send across the Internet.
2. Active attacks: that is, when a hacker is trying to enter your system.

Some companies may be tempted to ignore the Internet, but for others it can become a competitive advantage. Some of the advantages to open and unlimited communications are increased marketing exposure, increased productivity, and keeping current information available to others.

When you choose to connect your network to the Internet, one of the first steps is to ensure a security policy is in place and is followed. If you do not know what you have to loose, you do not know what to protect. Some guidelines for generating a security policy might be:

- How likely are threats?

- How important is the resource that needs protection?
- Which groups do you need to protect the resources from?
- What resources are you trying to protect in your policy?
- What is the cost trade-off?
- Make sure that your policy states when it needs to be reviewed.

10.2 AS/400 Application Security

One of the *features* of TCP/IP is that most (all?) of the security is the responsibility of the application. That means, for example, that it is up to Telnet or FTP to provide all of the security that is necessary for the system.

NOTE

Object authority is always your first line of defense. If you do not have a good plan for protecting your objects, your system is defenseless.

10.2.1 Telnet Security

Using the AS/400 Telnet server application on your system opens the door to your system through a Signon display. If the Telnet server is a critical part of your business operations, be aware that USERIDs and passwords flow ″in the clear″ across your network, your intranet, or the Internet. Hackers on the Internet can use Sniffers (line-tracing equipment) to access your logon passwords. Another threat is Spoofing, where the hacker′s equipment pretends to be your system by using its same IP address. This way, the hacker has users trying to sign on to the hacker′s system, typing in their USERID and password while the hacker records them.

What can you do if you must use the Telnet server over the Internet? Several things.

- Consider using a Firewall. Firewalls protect a secure network from an untrusted network. Firewalls can control traffic by packet filtering and by disabling routes. They also log and monitor traffic. Most routers offer some level of protection if configured properly.
- Give Telnet clients as little user authority as possible.
- Although the QMAXSIGN system value applies to Telnet, you reduce the effectiveness of this system value if you set up your system to configure virtual devices automatically. When the QAUTOVRT system value has a value greater than 0, the unsuccessful Telnet user can reconnect and attach to a newly-created virtual device. This can continue until one of the following occurs:
  1. All virtual devices are disabled, and the system has exceeded the limit for creating new virtual devices.
  2. All user profiles are disabled.
  3. The hacker succeeds in signing on to your system.
- You can use the QLMTSECOFR system value to restrict users with *ALLOBJ or *SERVICE special authority. The user or QSECOFR must be explicitly authorized to a device to sign on. Thus, you can prevent anyone with

*ALLOBJ special authority from using Telnet to access your system by ensuring that QSECOFR does not have authority to any virtual devices.

- Use system values that promote non-trivial passwords. For example:

  QPWDEXPITV Password expiration interval.

  QPWDLMTAJC Limit adjacent digits in password.

  QPWDLMTCHR Limit characters in a password.

  QPWDLMTREP Limit repeating characters in a password.

  QPWDMAXLEN Maximum password length.

  QPWDMINLEN Minimum password length.

  QPWDPOSDIF Limit password character positions.

  QPWDRQDDGT Require digit in password.

  QPWDRQDDIF Duplicate password control.

  QPWDVLDPGM Password validation program.

Remember that the most secure computer is one that's turned off. So if you *do not* want the Telnet server to run on your system, do the following:

1. Type CHGTELNA AUTOSTART(*NO)
2. Make sure that the public authority for the Start TCP/IP Server (STRTCPSVR) command is *EXCLUDE.
3. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for Telnet, do the following:
   - Type GO CFGTCP to display the Configure TCP/IP menu.
   - Select option 4 (Work with TCP/IP port restrictions).
   - On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
   - For the lower port range, specify 23.
   - For the upper port range, specify *ONLY.
   - For the protocol, specify *TCP.
   - For the user profile field, specify a user profile name that is protected on you system (not shared or adopted). By restricting the port to a specific user, you automatically exclude all other users.

The port restrictions take effect the next time you start TCP/IP. If TCP/IP is active when you change the port restrictions, you should end TCP/IP and start it again.

Make note of these changes in case you decide in the future that you want to remove these restrictions.

10.2.2 FTP Security

If you want to allow FTP clients to access your system, be aware of the following issues:

- Your object authority scheme may not provide detailed enough protection when you allow FTP on your system. For example, when a user has authority to view a file (*USE authority), the user can also copy the file to a

PC or to another system. You may want to protect some files from being copied to another system.

Beginning with V3R2, FTP exit programs are available to restrict the FTP operations that users can perform. You can use the FTP Request Validation Exit to control what operations you allow. For example, you can reject GET requests for specific database files. You can use the FTP Server Logon Exit to authenticate users who log on to the FTP server. An FTP exit program also defines an anonymous FTP user profile. Within this profile, you can limit the storage size allotted for an anonymous FTP user, thereby reducing the impact of being ″bombed″ by hackers. For an example of an FTP exit program, please see 6.1.4, “FTP Exit Programs” on page 112. Also see ″TCP/IP User Exits″ in the *TCP/IP Configuration and Reference* book for more information on FTP exit programs.

- As with Telnet, FTP passwords flow ″in the clear″ between the client and the server. Your FTP application may be vulnerable to sniffing.
- The QMAXSIGN system value applies to Telnet but not to FTP. With FTP, the system breaks the connection after five unsuccessful sign-on attempts, but the user can simply establish a new connection. Theoretically, an FTP user has ***unlimited*** attempts to break into your system.

  The system writes message CPF2234 to the QHST log for each unsuccessful attempt. You can write a program to monitor the QHST log for the message. If the program detected repeated unsuccessful attempts, it can end the FTP server.
- If you plan to use FTP batch support to transfer files between systems, remember that the program must send both user ID and password to the server system. Either the user ID and password must be coded in the program, or the program must retrieve them from a file. Both of these options are a potential security exposure. You must use object security to protect the USERID and password information. You should also use a single USERID that has very limited authority on the target system.
- FTP provides remote-command capability, just as APPC and AS/400 Client Access do. The Remote Command (RCMD) FTP-server subcommand is the equivalent of having a command line on the system. Before you allow FTP, you must ensure that your object security scheme is adequate. FTP exit points can be used to restrict the use of the RCMD subcommand.

If you *do not* want the FTP server to run on your system, do the following:

1. Type `CHGFTPA AUTOSTART(*NO)`.
2. Make sure that the public authority for the Start TCP/IP Server (STRTCPSVR) command is *EXCLUDE.
3. To prevent someone from associating a user application, such as a socket application, with the port that the system normally uses for FTP, do the following:
   - Type `GO CFGTCP` to display the Configure TCP/IP menu.
   - Select option 4 (Work with TCP/IP port restrictions).
   - On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
   - For the lower port range, specify 20.
   - For the upper port range, specify 21.

- For the protocol, specify *TCP.
- For the user profile field, specify a user profile name that is protected on you system (not shared or adopted). By restricting the port to a specific user, you automatically exclude all other users.

Again, port restrictions take effect the next time you start TCP/IP. If TCP/IP is active when you change the port restrictions, you should end TCP/IP and start it again.

Make note of these changes in case you decide in the future that you want to remove these restrictions.

10.2.3 HTTP Security

The HTTP server provides HTML documents to World Wide Web browser clients. It can also allow them to request functions that run AS/400 programs on their behalf. The HTTP server uses the CGI specification to call these programs. HTTP bypasses the normal AS/400 Signon. However, the HTTP administrator on the AS/400 system must explicitly authorize all CGI programs through the HTTP configuration.

The primary purpose of the HTTP server is to provide access to visitors through your intranet or the Internet and World Wide Web to your AS/400 system. While maintaining the tightest security possible, you need to ensure that security practices do not negatively impact the value of your WEB site. On the other hand, you need to ensure that HTTP activity does not jeopardize the integrity of your system or your network. Consider the following:

- A client cannot do anything with the HTTP server until the server administrator defines directives for the server. To define directives, you must use the Work with HTTP Configuration (WRKHTTPCFG) command. This command requires *IOSYSCFG special authority.
- Every request to the HTTP server must pass two security checks:
  1. The request must be defined explicitly in the directives for the HTTP server through Pass, Fail, Exec, and Enable statements, just to name a few.
  2. The QTMHHTTP user profile or *PUBLIC must have authority to the system resources for the request. All HTTP server activity runs under the QTMHHTTP user profile.
- When you add a directive to the HTTP server, make the template value for the path as specific as possible. This reduces the chance that someone can browse through your system and discover files. Avoid using generic file names and wildcards.
- Keep in mind that a FAIL statement in an HTTP configuration is case sensitive while most of the AS/400 file systems are case in-sensitive. The Fail statement is actually not too useful. The default action (if the URL does not match any Pass or Exec statements) of the HTTP server is to fail. So, code your Pass and Exec statements very carefully to allow only those operations in the directories as needed. Do not count on the Fail directive primarily. For example, read the following two lines in an HTTP configuration.

  ```
  Fail /QSYS.LIB/PAYROLL.LIB/*
  Fail /qsys.lib/payroll.lib/*
  ```

All a hacker has to do is issue the following URL, alternating in upper and lower case, and they are in the payroll library.

```
http://systemname/QSyS.lIb/PaYrOlL.lIb/*
```

So you see this example works as long as a Pass statement inadvertently allowed it. It is a better practice to carefully allow access to AS/400 objects on a selective basis rather than allowing access to all objects and then blocking only selected objects.

- Use Map or Pass directives to mask the file names on your AS/400 web server. For example, the client (browser) might issue a URL that looks similar to the following:

  ```
  http://hostname/webdata/products
  ```

  Use the WRKHTTPCFG command to add a Pass directive to the HTTP server configuration that looks similar to the following:

  ```
  Pass /webdata/products
  /QSYS.LIB/WEBDATA.LIB/WWWDATA.FILE/PRODUCTS.MBR
  ```

  The requester who sees this URL has no idea that the product data is in the WWWDATA file in the WEBDATA library on your AS/400 system. This method protects (hides) your AS/400 file names and library names from potential hackers. It also gives you the flexibility to change your AS/400 application without having to change the URL.

- Some HTTP server requests need to run an AS/400 program, for example, to access data on your system. Before the program can run, the server administrator must map the request URL to a specific user-defined program that conforms to CGI user-interface standards.

- HTTP provides read-only access to your AS/400 system. HTTP server requests cannot update or delete data on your system with one exception! You can enable the DB2WWW CGI program to access your AS/400 database. The system uses a script (which is similar to an exit program) to evaluate the requests to the DB2WWW program. Therefore, the system administrator can control what actions the DB2WWW program can take.

- The HTTP server provides an access log that you can use to monitor both accesses and attempted accesses through the server.

- For more information on the HTTP server configuration, see 8.2.2, “HTTP Server Directives” on page 166. Also see the HTTP chapter of the *TCP/IP Configuration and Reference*, SC41-3402.

10.2.4 Workstation Gateway Security

The Workstation Gateway server (WSG) provides a TCP/IP application that transforms AS/400 5250 applications into HTML for dynamic display on Web browsers. When you set up the Workstation Gateway server, you control whether users see a Signon display or whether an exit program handles Signon.

From a security standpoint, you need to understand the purposes for the WSG server on your system and the client environments that use it. For example, some users of the WSG server might be travelling employees who are using an intranet and accessing your system from inside a firewall. Other users might be visitors to your WEB site who want to request additional information. We do not recommend implementing the WSG without an exit program in place in those situations where your system is not sufficiently protected by a firewall. Think of the WSG as potential Telnet access to your system to anyone with a Web

browser on the Internet. For an example of a WSG exit program, please see 9.3.4, “5250 HTML Workstation Gateway Application Logon Exit Program” on page 252.

Users can access the WSG server in these ways:

- From a direct request by the client browser.
- From an indirect request when the HTTP server gives control to the WSG server.
- From a specific HTTP connect request. For example, A WEB site visitor might select an area of your WEB site that says, ″Send me additional information″. The HTTP server can send a request to the WSG server to show a display that asks for the name and mailing address of the visitor.

The following are security considerations when you allow the WSG server to run on your system.

- To configure the WSG server, you use the Change Workstation Gateway Attributes (CHGWSGA) command. One configuration controls all of the WSG sessions on your system. You can specify the following values:

  Inactivity timeout: The WSG server does not use the QINACTIV system value. The WSG server has its own value for determining how long the system waits before it ends an inactive session.

  Display sign on panel: When a WSG request comes from a World Wide Web browser, this value controls whether the system sends your system-defined Signon display. You should use a WSG logon exit program to perform user validation.

  Often, a request to the WSG server is to perform a specific, limited function, such as presenting a form for completion. The Signon display is not necessary in this environment. In fact, the Signon display provides an opportunity for ″hacking″ that is not available when your exit program bypasses Signon and uses a user profile with very limited authority.

  Access logging: You may find it useful to have your system keep a record of the accesses to your WSG server, particularly when you provide a new application.

- As with Telnet, you can use the QLMTSECOFR system value as one method to limit the capability of WSG users. When the QLMTSECOFR value is 1, you can prevent any user with *ALLOBJ special authority from signing on to a WSG workstation unless the user or QSECOFR is explicitly authorized.

  If you *do not* want anyone to use WSG to access your system, you can prevent the WSG server from running by doing the following:

1. Type CHGWSGA AUTOSTART(*NO). The system is shipped with *NO as the default value.
2. Make sure that the public authority for the Start TCP/IP Server (STRTCPSVR) command is *EXCLUDE.
3. To prevent WSG from starting and to prevent someone from associating a user application such as a socket application with the port that the system normally uses for HTTP, do the following:
   - Type GO CFGTCP to display the Configure TCP/IP menu.
   - Select option 4 (Work with TCP/IP port restrictions).

- On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
- For the lower port range, specify 5061.
- For the upper port range, specify *ONLY.
- For the protocol, specify *TCP.
- For the user profile field, specify a user profile name that is protected on you system (not shared or adopted). By restricting the port to a specific user, you automatically exclude all other users.

Again, port restrictions take effect the next time you start TCP/IP. If TCP/IP is active when you change the port restrictions, you should end TCP/IP and start it again.

Make note of these changes in case you decide in the future that you want to remove these restrictions.

10.2.5 SMTP Security

SMTP (Simple Mail Transfer Protocol) provides the capability to distribute documents to your system. The system does not perform any Signon processing for SMTP. If you want to allow SMTP clients to access your system, be aware of the following security issues:

- If possible, avoid using an *ANY *ANY entry in the system distribution directory. When your system does not have an *ANY *ANY entry, it is more difficult for someone to attempt to use SMTP to flood your system. *Flooding* or *Mail Bombing* occurs when your auxiliary storage is filled up with unwanted mail that is being routed through your system.
- To prevent a user from swamping your system with unwanted objects by flooding or mail bombing, be sure that you have set adequate threshold limits for your auxiliary storage pools (ASPs).

If you *do not* want anyone to use SMTP to distribute mail to your system, you should prevent the SMTP server from running. Do the following:

1. To prevent SMTP server jobs from starting automatically when you start TCP/IP, type:

   ```
   CHGSMTPA AUTOSTART(*NO)
   ```

2. Make sure that the public authority for the Start TCP/IP Server (STRTCPSVR) command is *EXCLUDE, which is the shipped value.
3. To prevent SMTP from starting and to prevent someone from associating a user application such as a socket application with the port that the system normally uses for SMTP, do the following:
   - Type GO CFGTCP to display the Configure TCP/IP menu.
   - Select option 4 (Work with TCP/IP port restrictions).
   - On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
   - For the lower port range, specify 25.
   - For the upper port range, specify *ONLY.
   - For protocol, specify *TCP.
   - For the user profile field, specify a user profile name that is protected on your system. A protected user profile is a user profile that does not own programs that adopt authority and do not have a password that is known

by other users. By restricting the port to a specific user, you automatically exclude all other users.

- Repeat the previous steps for the *UDP protocol.

4. To provide extra protection, hold the SNADS distribution queues that the SMTP application uses by typing in the following commands:

```
HLDDSTQ DSTQ(QSMTPQ) PTY(*NORMAL)
HLDDSTQ DSTQ(QSMTPQ) PTY(*HIGH)
```

Again, port restrictions take effect the next time you start TCP/IP. If TCP/IP is active when you change the port restrictions, you should end TCP/IP and start it again.

Make note of these changes in case you decide in the future that you want to remove these restrictions.

10.2.6 POP Security

The POP (Post Office Protocol) server provides a simple store-and-forward mail system. The POP server holds mail temporarily until a mail client retrieves it. The client/server interface of the POP server requires the services of the SMTP server. A mail client must have a user ID and a password to retrieve mail from the POP server.

If you want to allow POP clients to access your system, be aware of the following security issues:

- The POP mail server provides authentication for clients who attempt to access their mailboxes. The client sends a USERID and password to the server. The POP server uses a system API to verify the USERID and password against the AS/400 user profile for that user.

  Because you do not have control over how the USERID and passwords are stored on the POP client, you might want to create a special user profile that has very limited authority on you system. To prevent anyone from using the user profile for an interactive session, you can set the following values in the user profile:

  Set initial menu (INLMNU) to *SIGNOFF.

  Set initial program (INLPGM) to *NONE.

- To prevent a user from swamping your system with unwanted objects or mail bombing, be sure that you have set adequate threshold limits for your auxiliary storage pools (ASPs).

If you *do not* want anyone to use POP to access your system, you should prevent the POP server from running. Do the following:

1. To prevent POP server jobs from starting automatically when you start TCP/IP, type:

   ```
   CHGPOPA AUTOSTART(*NO)
   ```

2. Make sure that the public authority for the Start TCP/IP Server (STRTCPSVR) command is *EXCLUDE, which is the shipped value.

3. To prevent POP from starting and to prevent someone from associating a user application such as a socket application with the port that the system normally uses for POP, do the following:

- Type GO CFGTCP to display the Configure TCP/IP menu.
- Select option 4 (Work with TCP/IP port restrictions).
- On the Work with TCP/IP Port Restrictions display, specify option 1 (Add).
- For the lower port range, specify 110.
- For the upper port range, specify *ONLY.
- For protocol, specify *TCP.
- For the user profile field, specify a user profile name that is protected on your system. A protected user profile is a user profile that does not own programs that adopt authority and does not have a password that is known by other users. By restricting the port to a specific user, you automatically exclude all other users.

Again, port restrictions take effect the next time you start TCP/IP. If TCP/IP is active when you change the port restrictions, you should end TCP/IP and start it again.

Make note of these changes in case you decide in the future that you want to remove these restrictions.

10.2.6.1 SLIP Security

AS/400 TCP/IP support includes Serial Interface Line Protocol (SLIP). SLIP provides low-cost point-to-point connectivity. A SLIP user can connect to a LAN or a WAN by establishing a point-to-point connection with a system that is part of the LAN or WAN.

SLIP runs on an asynchronous connection. You can use SLIP for dial-up connection to and from an AS/400 system. No SLIP configuration exists on your system when it ships. Therefore, if you do not want SLIP (and dial-up TCP/IP) to run on your system, do not configure any profiles for SLIP.

If you want SLIP to run on your system, you must create one or more SLIP (point-to-point) configuration profiles using the Work TCP Point-to-point (WRKTCPPTP) command. The topics that follow discuss how you can set up security for SLIP configuration profiles. A **user profile** is an AS/400 object that allows a Signon. Every AS/400 job must have a user profile to run. A **configuration profile** stores information that is used to establish a SLIP connection with an AS/400 system. When you start a SLIP connection to an AS/400 system, you are simply establishing a link. You have not yet signed on and started an AS/400 job. Therefore, you do not necessarily need an AS/400 user profile to start a SLIP connection to an AS/400 system as in Client Access/400. However, as you see in the discussions that follow, the SLIP configuration profile may require an AS/400 user profile to determine whether to allow the connection.

Securing Dial-In SLIP Connections: If you want to validate systems that dial in to your system, then you want the requesting system to send a USERID and password. Your system can then verify the USERID and password. If the USERID and password are not valid, your system can reject the session request.

To set up dial-in validation, do the following:

1. Create a user profile that the requesting system can use to establish the connection. The USERID and password that the requester sends must match this user profile name and password.

   For additional protection, you probably want to create user profiles specifically for establishing SLIP connections. The user profiles should have limited authority on the system. If you do not plan to use the profiles for any function except SLIP connections, you can set the following values:

   An initial menu (INLMNU) of *SIGNOFF.

   An initial program (INLPGM) of *NONE.

   These values prevent anyone from signing on interactively with the user profile.

2. Create an authorization list for the system to check when a requester tries to establish a SLIP connection.

3. Use the Add Authorization Entry (ADDAUTLE) command to add the user profile that you created in the first step to the authorization list. You can create a unique authorization list for each point-to-point configuration profile, or you can create an authorization list that several configuration profiles share.

4. Use the WRKTCPPTP command to set up a TCP/IP point-to-point *ANS profile that has the following characteristics:

   - The configuration profile must use a connection dialog script that includes the user-validation function. User validation includes accepting a USERID and password from the requester and validating them. The system ships with several sample dialog scripts that provide this function. Keep in mind that the AS/400 dialog script that is chosen must match the remote user's dialog script.
   - The configuration profile must specify the name of the authorization list that you created in the second step. The USERID that the connection dialog script receives must be in the authorization list.

Because of the different security practices and capabilities of your communication partners, you might want to create different configuration profiles for different requesting environments. Use the STRTCPPTP command to set your system up to accept a session for a specific configuration profile. You can start sessions for some configuration profiles only at certain times of the day, for example. You might use security auditing to log the activity for the associated user profiles.

Securing Dial-Out SLIP Connections: Users on your AS/400 system might want to establish dial-out connections to systems that require user validation. The connection dialog script on your AS/400 system must send a USERID and a password to the remote system. The AS/400 system provides a secure method for storing that password. Even though the system stores the connection password in encrypted form, it decrypts the password before sending it. SLIP passwords, such as Telnet and FTP passwords are sent ″in the clear″. However, unlike Telnet and FTP, the SLIP password is sent before the systems establish TCP/IP mode.

The default file for storing SLIP connection dialog scripts is QUSRSYS/QATOCPPSCR. The public authority for this file is *USE, which prevents public users from changing the default connection dialog scripts.

When you create a connection profile for a remote session that requires validation, do the following:

1. Ensure that the Retain Server Security Data (QRETSVRSEC) system value is 1 (YES). This system value determines whether you allow passwords that can be decrypted to be stored in a protected area on your system.
2. Use the WRKTCPPTP command to create a configuration profile that has the following characteristics:
   - For the mode of the configuration profile, specify *DIAL.
   - For the Remote service access name, specify the USERID that the remote system expects. For example, if you are connecting to another AS/400 system, specify the user profile name on that AS/400 system.
   - For the Remote service access password, specify the password that the remote system expects for this USERID. On your AS/400 system, this password is stored in a protected area in a form that can be decrypted. The names and passwords that you assign for configuration profiles are associated with the QTCP user profile. The names and passwords are not accessible with any user commands or interfaces. Only registered system programs can access this password information.
   - For the connection dialog script, specify a script that sends the USERID and password . The system ships with several sample dialog scripts that provide this function. When the system runs the script, the system retrieves the password, decrypts it, and sends it to the remote system.

For more information on SLIP dialog scripts, see 3.2, “SLIP Dial-Up Support on the AS/400” on page 28. Also see *TCP/IP Configuration and Reference*, SC41-3420, for more information on how to configure and set up SLIP.

This chapter describes some of the features provided by OS/400 TCP/IP applications in the area of security. Each of the TCP/IP applications, because they are the ones to actually provide security, are also addressed in each chapter. For more information about security with each of the applications, please see:

Application	Please see...
Telnet	5.3, “Telnet Security” on page 100.
FTP	6.2.4, “FTP Server Security” on page 118.
WWW server	8.4.3, “Logging the Access of the Web server” on page 188.
Workstation Gateway	9.3.4, “5250 HTML Workstation Gateway Application Logon Exit Program” on page 252.
SLIP	3.2, “SLIP Dial-Up Support on the AS/400” on page 28.

10.3 AS/400 Security Features

The AS/400 system, with its inherent built in security, makes an excellent candidate to place on the Internet. The AS/400 system is not susceptible to the security holes commonly found in UNIX systems.

NOTE

Object authority is always your first line of defense. If you do not have a good plan for protecting your objects, your system is defenseless.

Some examples of the security features on the AS/400 system are:

- Object Authority

  The most common examples of objects are files and programs. Other types of objects include commands, queues, libraries, and folders. Each object on the system can be secured.

- System level security (10-50)

The AS/400 system offers five levels of security:

Level 10 -- Physical Security Only
Level 20 -- Password Security
Level 30 -- Password and Resource Security
Level 40 -- Integrity Protection (recommended)
Level 50 -- Enhanced Integrity Protection

- Authentication procedures (user profiles + password + Environment):

  Use of user profiles to limit administration capability.
  Ability to force user password differences as well as timing.
  Capability to set user environment.

- System directory:
- Limited access through servers:

  The system administrator may disallow any functions including the common TCP/IP servers such as Telnet and FTP as well as the AS/400 commands.

With the preceding security features, the AS/400 system can have, in relation to Web Serving, support for:

1. Read-only Bulletin Boards
2. Limited access to files and execution of applications

More specific information on AS/400 security is found in the *Basic Security Guide*, SC41-3301, and *Security - Reference*, SC41-3302.

10.4 Security Standards on the Internet

To provide continuity and consistency for accessing data on the Internet, many security standards are being developed. Standards provide the framework for client-to-server or end-to-end security. These standards are essential for providing secure communications for such things as banking and electronic commerce. The standards provide a framework for client-to-server or end-to-end security.

Security standards can be divided broadly into two categories:

1. Access security
2. Transaction security

10.4.1 Access Security

Access security is the ability to secure your AS/400 files and programs on an individual basis. There are several approaches to securing your AS/400 system. A few examples might be by user, by user group, or even by terminal location. All AS/400 administrators should have a security policy for anyone accessing their systems.

10.4.2 Transaction Security

The Internet does not provide inherently secure communications between Web browsers and Web servers. Often, this lack of security is no cause for concern, but some applications demand high security. Some examples of these might be:

- Buying and selling transactions
- Authentication processes
- Message integrity
- Transactions that involve receipts and signatures

Web browsers and Web servers can communicate securely, but only with security enhancements to both browser and server. The browser and server must both use the same enhanced communications protocol; otherwise, they cannot communicate.

The two emerging standards as explained in Internet drafts are:

1. Secure Sockets Layer (SSL):

   The SSL Protocol is application protocol independent, that is, a higher level application protocol such as HTTP, FTP,and Telnet can layer on top of the SSL Protocol transparently. The SSL Protocol can negotiate an encryption algorithm and session key as well as authenticate a server before the application protocol transmits or receives its first byte of data. All of the application protocol data is sent encrypted, ensuring privacy.

   The SSL protocol provides channel security and has three basic properties.

   - The channel is private. Encryption is used for all messages after a simple handshake is used to define a secret key.
   - The channel is authenticated. The server end of the conversation is always authenticated, while the client end is optionally authenticated.
   - The channel is reliable. The message transport includes a message integrity check (using a MAC).

2. Secure HyperText Transfer Protocol (SHTTP):

   Secure HTTP is an extension of HTTP that provides additional security measures for the Web. Authentication allows a client to ensure it is communicating with the proper server, and a server to ensure it is communicating with the proper client. Encryption makes the data sent over the Internet appear unintelligible to anyone eavesdropping. Digital signatures provide two benefits. First, they verify that data sent on the network was not changed, and second, the receiver of the data can prove that the sender actually sent the data.

10.5 Firewalls

Firewalls are a hardware and software implementation that defines, controls, and limits the access to internal or secure networks from outside or unsecured networks such as the Internet. Firewalls control network traffic and access, provide privacy, and log access and intrusion attempts. Some routers provide a certain level of protection by themselves but should not be considered a substitute for true firewall protection.

Examples of two popular types of firewalls are:

- Packet Filter:

  This is the most common strategy used to separate private IP networks from the Internet. It is accomplished by inserting a router between the internal and external networks. This router filters all IP packets routed through it based on IP address. This firewall implementation is controlled by filter rules. For example, it can be configured to permit Telnet from 192.168.*.* to *.*.*.* and also permit HTTP from *.*.*.* to 192.168.67.3. Proper configuration allows seamless and transparent access to trusted users.

- Bastion:

  A Bastion firewall is a machine in which IP forwarding is turned off and is placed between the secure and unsecured networks. Since the routing function is turned off, the only place you can access both networks is the bastion machine itself. Therefore, only users with a logon to the bastion machine may pass through to the other network.

Additional functions that firewalls may provide are Proxy and SOCKS servers.

An example of the proxy function is much the same as the bastion configuration in which an AS/400 user might Telnet to the bastion, authenticate itself, and then use the Telnet application on the bastion machine. In the case of World Wide Web access, the end user configures a WWW client to point to the bastion machine, it runs a WWW server, and the bastion machine does the actual looking up of the WWW URL.

SOCKS is an emerging standard for an application-level gateway that does not require the overhead of a proxy server. The SOCKS server results in a similar bastion configuration, since the session is broken at the firewall. The difference is that the user does not need to perform the double login. SOCKS needs to have new versions of the client code (called SOCKSified clients) and have a separate configuration on the firewall. At present, the AS/400 system does not have any SOCKSified clients. Some clients are available for WWW access such as OS/2′s WebExplorer and NetScape′s client. There are limited FTP and Telnet SOCKSified clients available.

More specific information on AS/400 security is found in the *Basic Security Guide*, SC41-3301, and *Security - Reference*, SC41-3302.

10.5.1 IBM Secure Network Gateway

IBM has in its family of Internet Connection products, the enhanced Secured Network Gateway for AIX. The Internet Connection Secured Network Gateway is a firewall that gives you a secure internal network that can have limited exchanges with users outside in the Internet, but is selective in who it allows in. It isolates your internal network from other networks in the Internet and provides typical TCP/IP applications such as (FTP and Telnet) that access hosts outside the secure network by protecting the secure network. The IBM firewall acts as a barrier between your network and the rest of the Internet.

In order to control access between your network and the Internet and facilitate authorized transactions, the Internet Connection Secured Gateway program provides these services and barriers:

1. Proxy Server is a gateway from one network to another for a specific network application. The Internet Connection Secured Network Gateway provides proxy servers for such key operations as Telnet emulators and file transfer programs.
2. SOCKS Servers is a SOCKS server that provides a remote application program interface so that the functions executed by client programs in secure domains are piped through secure servers at the firewall workstations. The SOCKS server verifies the user and redirects the function through the firewall.
3. Secure IP Tunnels uses the IBM-defined protocol. The Secured Network Gateway provides a secure IP ″tunnel″ from your secure network, through the Internet, and to another secure network that also has a secure gateway on the other end of the tunnel.
4. Filters provide ways to limit user access into or out of a secure network based on such things as the origin or destination Internet address or TCP/IP protocol.
5. Domain Name Service prevents users outside the secure network from seeing addresses of secure hosts, while assisting secure hosts in resolving addresses of hosts in the non-secure network.
6. Mail Handler provides mail routing from outside the firewall to a secure mail server within your network, and from your network out through the firewall mail handler.

10.5.2 AS/400 Internet Security Scenario

Some common considerations when connecting your AS/400 system might be as follows:

- Use a firewall if you intend to connect your production system to the Internet.
- Do not start or use unneeded servers such as FTP or Telnet.
- Restrict commands. Some examples are:
  - All restore commands (RST...)
  - Start PassThru (STRPASTHR)
  - Start TCP Server (STRTCPSVR)
  - Power Down System (PWRDWNSYS)
- Have no compilers on your system, or severely restrict access to them. Since AS/400 program objects can only be created by restore or compile,

these two functions should be tightly controlled on your AS/400 system that is connected to the Internet. Eliminating the ability to either restore or compile reduces the possibility of a hacker installing their own rogue program virus.

- Enforce password restrictions with system values.
- Change public authority to *EXCLUDE.
- Check that IP datagram forwarding is not allowed. Type the CHGTCPA IPDTGFWD (*NO) command. This restricts access to your LAN from users coming in through the Internet.
- If this AS/400 system is a gateway system for an SNA network (that is, TCP/IP to the Internet on one side and a Network Node to an SNA APPN network on the other), then do not turn on AnyNet/400 (through the CHGNETA ALWANYNET(*NO) command). If you are using AnyNet/400 to carry SNA data across the Internet (which is a fine thing to do) understand that the really smart hacker (who understands the topology of your SNA network) can STRPASTHR to any system inside of your SNA network without first needing to login to your gateway AS/400 system. This is due to the fact that AnyNet/400 (because you configured it to) treats the incoming data the same as an APPN intermediate node connection.

  This point just reinforces the fact that you must consider the security of *all* of your systems in a network, not just the ones connected to the external network.

10.5.3 Security Service Offering

IBM AS/400 Availability Services offers a complete set of startup, installation, and consulting services to integrate your company into network centric computing quickly. IBM offers the following service to evaluate OS/400 security as it relates to the Internet.

AS/400 Security Review (U.S. only): IBM specialists review your system security with emphasis on Internet exposure. This offering includes assessment of your exposures, a plan to implement security measures, and implementation of that plan.

NOTE

For more information on AS/400 security, see *Tips and Tools for Securing Your AS/400*, SC41-3300.

Appendix A. Installing the ITSO Company Demo

These instructions lead you step-by-step through the process of restoring and configuring the ITSO Company Application and a three-day class on the V3R2 Internet Connection for AS/400.

These instructions do not cover TCP/IP configuration and assume that you have a working TCP/IP connection between your AS/400 system and a PC. We further assume that the PC is capable of reading the electronic media that is included with this redbook. Further more, we assume that you can do basic TCP/IP tasks such as copying a file between the PC and the AS/400 system through FTP.

A.1.1 Overview of the Install and Configuration Process

The steps necessary to restore, configure, and run the ITSO Company Application and the three day class are as follows:

Overview **Description**

Restore You need to transfer the AS/400 save files (SAVF), HTML, and GIFs from the PC compatible CD-ROM that came with this redbook to your V3R2 AS/400 system. Most likely, this is accomplished by using FTP to transfer the data to the AS/400 system and, in fact, this is how we lead you through this process.

Other options exist for the savvy. Once easy one requires Client Access/400 on the PC to give you a virtual drive view of the entire AS/400 file system. Then the copy from the CD-ROM drive of the PC to the IFS on the AS/400 system is as simple as a drag-and-drop. Note that the two AS/400 save files that are on the CD-ROM are in binary PC format and they still must be copied to an AS/400 SAVF through FTP to reconstruct the data and format. It does not matter, however, if the binary save file originates from the CD-ROM or the AS/400 IFS.

Configure You need to configure a number of things on the AS/400 system to allow it to become a World Wide Web application server. We have broken this into two pieces.

- Automatic Configuration: We have written a small CL program that can:
  - Copy a working HTTP configuration file to QUSRSYS/QATMHTTPC member CONFIG.
  - Add exit programs for FTP.
  - Add exit programs for WSG.
  - Create user profiles.
  - Start the HTTP server.

  Note: At this point, most of the class and ITSO Company Application work. Only a few of the URLs that need to be manually updated in the next step do not work at this time.
- Manual Configuration: Unfortunately, some of the URLs that we use in the class and the ITSO Company Application are

hard-coded. You must manually go in and change these before they work on your AS/400 server.

Running At this point you should be able to run both the three-day class and the ITSO Company Application.

Lab setup The three-day class has four hands-on labs built in. Setting up this student team environment takes some extra time and skill.

When you are done, you should have the following AS/400 libraries and directories on your system as shown in Figure 193.

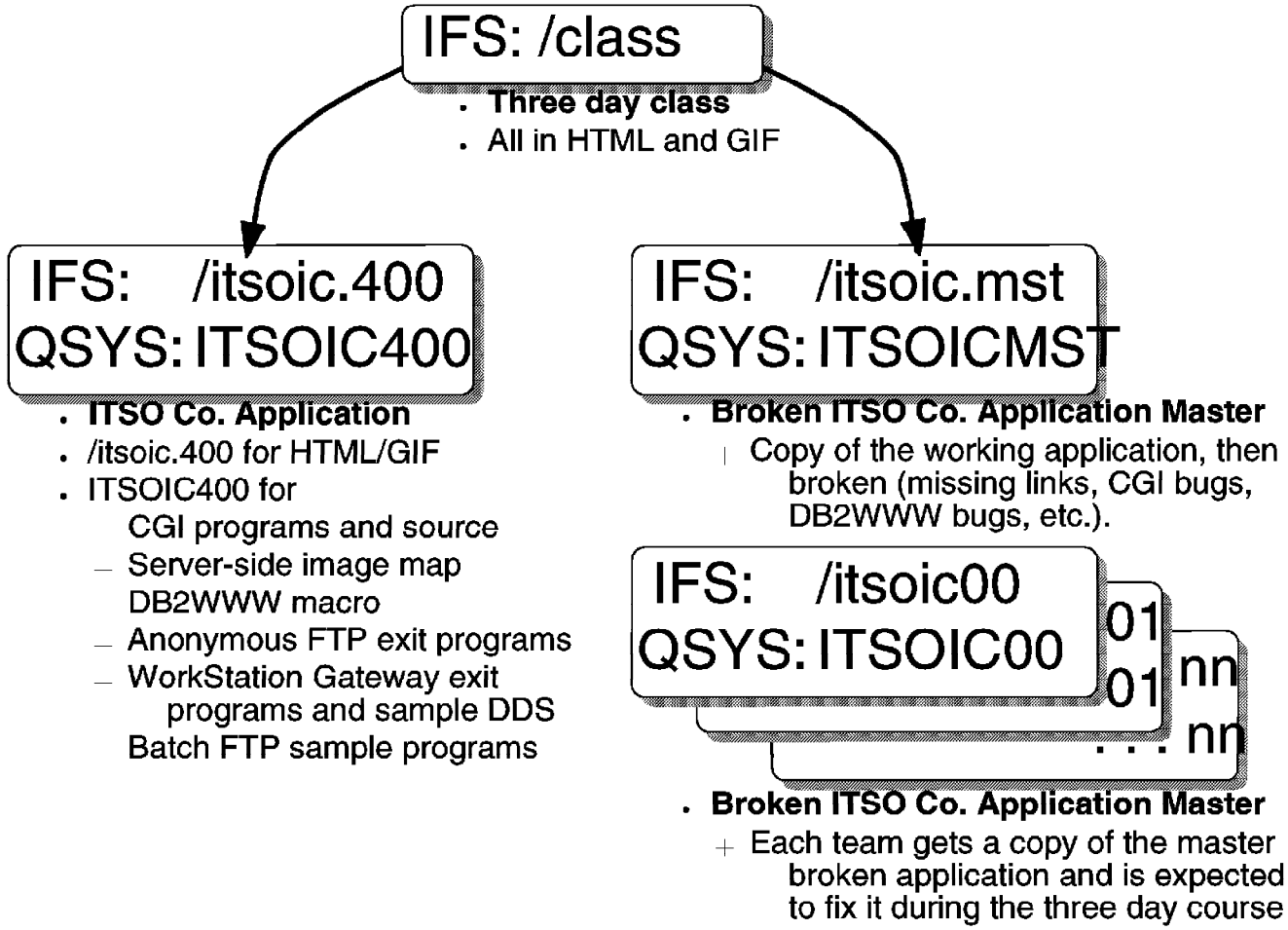


Figure 193. ITSO Company Application Libraries and Directories

A.1.2 Restore the ITSO Company Application and Class

The primary task is to transfer the HTML, GIF, and AS/400 save files from the electronic media to the AS/400 system. This redbook comes with CD-ROM that contains all the files in a format that can be read by most PCs. All the files that you need can be found in the following directory structure on the CD-ROM:

Directory	Contents
\CLASS	All the HTML and GIFs for a three-day class on the V3R2 Internet Connection for AS/400. The three-day class also makes use of the ITSO company application found in \ITSOIC.400 and \ITSOIC.MST (see the following list). These HTML and GIF files are copied to the AS/400 directory by the same name (/class).
\ITSOIC.400	All the HTML and GIFs to support the ITSO company application that is the source for most of the examples in this redbook. These HTML and GIF files are copied to the AS/400 directory by the same name (/itsoic.400).
\ITSOIC.MST	All the HTML and GIFs to support the broken ITSO company application that are used for the student′s hands-on labs. These HTML and GIF files are copied to the AS/400 directory by the same name (/itsoic.mst).
\FTPSAVFS\ITSOIC40	This is a binary image of an AS/400 save file that contains all the source and programs for the ITSO company application. The ITSO company application has sample CGI, DB2WWW macros, server side image maps, and so on, and this is the save file that contains the AS/400 library ITSOIC400 with all of the supporting code. This binary file is transferred to an AS/400 save file through FTP so it later can be restored using the RSTLIB command.
\FTPSAVFS\ITSOICMS	This is a binary image of an AS/400 save file that contains all of the source and programs for the broken ITSO company application. This binary file is transferred to an AS/400 save file through FTP so it later can be restored using the RSTLIB command.

Here are the steps needed to restore all of the directories and libraries on your V3R2 AS/400 system. Again, these steps assume you are using FTP and assumes that your client (in this case, most likely a PC) and the server (in this case, your AS/400 system) are both running TCP/IP and are properly configured. It also assumes that your PC is running OS/2, but should work with only slight modification to any of Microsoft Window′s products.

1. Start FTP connection to your AS/400 system and logon with a user profile that has the authority to create and restore libraries and directories. From your PC client′s command prompt:
   - `ftp youras400`
   - `Name: youruserprofile`
   - `Password: yourpassword`
2. When you are connected, change the AS/400 system′s naming format to the IFS naming format by the command:

- quote site namefmt 1

or

- site namefmt 1

depending on if your FTP client supports the site command.

3. Let's also get a few client FTP commands out of the way that can help us later:
   - binary (Transfer data without modification.)
   - prompt (To turn off prompting for the FTP client.)
   - quote time 9999 (To increase the AS/400 timeout.)
   - lcd x: (To change the FTP client's current directory to the CD-ROM reader. Replace the 'x' with the drive letter of your CD-ROM reader.)
4. Create the needed save files on the AS/400 system with the following commands:
   - quote rcmd CRTSAVF FILE(QGPL/ITSOIC400)
   - quote rcmd CRTSAVF FILE(QGPL/ITSOICMST)
5. Transfer the save files to the AS/400 system with the following commands:
   - lcd /ftpsavfs
   - put itsoic40 /qsys.lib/qgpl.lib/itsoic400.savf
   - put itsoicms /qsys.lib/qgpl.lib/itsoicmst.savf
6. Then create the needed directories on the AS/400 system with the following commands:
   - mkdir /class
   - mkdir /itsoic.400
   - mkdir /itsoic.mst
7. Transfer the files to the AS/400 system with the following commands:
   - cd /class
   - lcd /class
   - mput *.*
   - cd /itsoic.400
   - lcd /itsoic.400
   - mput *.*
   - cd /itsoic.mst
   - lcd /itsoic.mst
   - mput *.*
8. Sign on to the AS/400 system with a 5250 workstation and restore the AS/400 libraries with the following commands:
   - RSTLIB SAVLIB(ITSOIC400) DEV(*SAVF) SAVF(QGPL/ITSOIC400) MBROPT(*ALL) ALWOBJDIF(*ALL)
   - RSTLIB SAVLIB(ITSOICMST) DEV(*SAVF) SAVF(QGPL/ITSOICMST) MBROPT(*ALL) ALWOBJDIF(*ALL)

A.1.3 Automatic Configuration of ITSO Company Application and Class

We have written a small CL program that automates some of the configuration necessary to run both the ITSO Company Application and the three-day class.

Note: We strongly recommend that you do not run this program if you are in any way concerned about security. It, among other things, creates user profiles on your system that use the same password as the user name. This is, of course, a bad thing to do in general.

This program does the following:

- Copies a working HTTP configuration file to QUSRSYS/QATMHTTPC member CONFIG:

  It does this with the MBROPT of *ADD, which causes the least disruption of HTTP directives that you have already coded. This copy works best when the HTTP configuration file in QUSRSYS is the default shipped with the install tape of all comments.

  Note: If you have already started to make changes to the HTTP configuration directives, we suggest that you edit this file (with the WRKHTTPCFG command) to ensure this simple-minded install program has not messed up.
- Add exit programs for FTP:

  These exit programs allow an anonymous read-only FTP access to your AS/400 system.
- Add exit programs for WSG:

  This logon exit program causes the WSG to call a sample program (that has read-only access to a simple file in ITSOIC400 library) named SHTMLR. This happens when the Web client passes a URL that contains extra information.
- Create user profiles:

  By far the most dangerous thing that our ICINSTALL program does is to create a number of user profiles on the system. Again, if this is a production system, you might want to investigate first if this is going to pose any security problems on your system.

  ICINSTALL creates the following user profiles:
  - ANONYMOUS with password of *NONE
  - JOEGOOD with password of *USRPRF
  - JOENORMAL with password of *USRPRF
  - JOEBAD with password of *USRPRF

  These user profiles are used by different portions of the redbook and the ITSO Company Application to demonstrate, oddly enough, security.
- Enable the user profiles for HTTP:

  The user profiles QTMHHTP1 and QTMHHTTP are disabled as shipped with the TCP/IP Connectivity Utilities/400.
- Starting WSG, HTTP, and FTP servers:

  The ICINSTALL program changes the attributes of the Workstation Gateway, HTTP server, and FTP server to automatically start the next time TCP/IP is started on the system (with the STRTCP command). It also starts these servers.

To run the automatic configure program:

- ENDTCPSVR SERVER(*FTP *WSG *HTTP)
- ADDLIBLE ITSOIC400

- CALL ICINSTALL
- G (To accept the security risk and as-is nature of this program).

Note: At this point, most of the class and ITSO company application work. Only a few of the URLs that need to be manually updated in the next step do not work at this time. You can skip ahead to A.1.5, "Running the ITSO Company Application and the Three-Day Class" on page 291 if you are really eager to see the working class and demo.

A.1.4 Manual Configuration of Hard-Coded URLs

Sorry, but we had to hard-code some of the URLs in both the ITSO Company Application and the three-day class. In particular, the anonymous FTP and Work Station Gateway URLs needed to be hard-coded in several different spots. This section should help you find and identify all of those locations where you need to change our host name of:

INTERNUT.RCHLAND.IBM.COM

to yours. Here are most of the places you must manually edit the source HTML, DB2WWW macro script or Workstation Gateway DDS source to change the URL to point to your AS/400 system.

The DB2WWW macros and DDS source for the Workstation Gateway are all found in AS/400 source physical files. You can simply use STRSEU to edit this source.

Note: The HTML source files are kept in the Integrated File System (IFS) and cannot simply be edited by SEU or any other AS/400 utility. These are ASCII stream files. You have two choices on how to edit these files:

1. FTP them to your PC hard disk. Edit them and make the necessary changes. FTP them back to the AS/400 IFS.

   If you do this, we suggest that you create a master subdirectory (such as \class) on your PC in which *all* the HTML and GIFs are stored. When you need to make changes to any one HTML or GIF file you make this change on the master PC, then FTP the changed files back to the AS/400 IFS to be served by the AS/400 HTTP server.

2. Use an application such as Client Access/400 to allow a PC to mount the AS/400 IFS the same as a virtual drive. Then, edit the ASCII HTML file using PC editors that actually read, then write the changes back to the AS/400 IFS.

   This is the most elegant solution of the two.

Here are the files in which you should change the ITSO host name of INTERNUT to your AS/400 host:

- Directory /class:
  - INT020.HTM
  - FTP050.HTM
  - WSG060.HTM
  - HTL010.HTM
  - CGL010.HTM
  - DBL010.HTM
  - PFL010.HTM
- Library ITSOIC400 (optionally the same locations in ITSOICMST):
  - SRCF HTML, Member IMVH200

– SRCF QDDSSRC, Member SHTMLD

A.1.5 Running the ITSO Company Application and the Three-Day Class

Here are the instructions on how to run the ITSO company application and the three-day class.

1. Finally, start a web browser on your PC and point it to URL:

   - `http://YourAS400/itsoic400/welcome.htm`

   and start your journey to the AS/400 Web serving. (Substitute YourAS400 with the address or TCP/IP host name of your AS/400 system.)

   Or point your browser to URL:

   - `http://YourAS400/class/agenda.htm`

   to see a three-day class on Internet Connection for AS/400.

A.1.6 Three-Day Class Lab Setup

The three-day class has a series of labs built into it that require some additional setup and configuration. The following is a high-level view of the steps that you must take to run these lab exercises:

1. Replicate the master student library ITSOICMST and directory /itsoic.mst. We have written a CL program for you that makes this job fairly easy. It is called ICSETUP and it is found in:

   `library ITSOIC400 file QCLSRC`

   You must first edit this source file and change our ITSO system name of INTERNUT to your AS/400 host name. Adjust the length if necessary.

   ```
   DCL        VAR(&SYSTEM) TYPE(*CHAR) LEN(8) VALUE(INTERNUT)
   ```

   Then, recompile this CL program. We also have created a command front end to this program by the same name. To run this program, type ICSETUP and then prompt with PF4. You should see:

```
                          Setup IC/400 Lab (ICSETUP)

Type choices, press Enter.

Program name . . . . . .   *CREATE       *CREATE, *DELETE, *RESET
number of user teams . . . . . .    1      Number
```

Figure 194. Three Day Class Lab Setup Command

   From here, you may create, delete, or reset (reset actually does a delete, then create). We suggest that you test this program with an initial number of user teams, including one that creates two teams with suffixes of 00 and 01.

2. Edit the HTTP configuration file to add unique directives for each of the student teams. We strongly suggest that you first copy the QUSRSYS/QATMHTTPC.CONFIG file to a source physical file to edit with STRSEU if you are going to make numerous changes. When you are done, copy the updated source physical file member back to the HTTP CONFIG file with the *REPLACE member option.

   Find and then copy the following section of HTTP configuration directives:

```
#*****************************************************#
#** Start:HTTP Directives for Team 00 ****************#
#** Copy this for each team. Then change 00 to nn...
# This is your web application:
Pass /ITSOIC00/*
Pass /itsoic00/*
# This statement for image mapping
Map /cgi-bin/imagemap00/* /qsys.lib/qtcp.lib/qtmhimag.p   >
Pass /qsys.lib/itsoic00.lib/imagemap.file/*
# This statement for CGI programming
Exec  /BonusCGI00/* /QSYS.LIB/ITSOIC00.LIB/*
# This statement for DB2WWW
Map /cgi-bin/db2www00/* /QSYS.LIB/QTCP.LIB/DB2WWW.PGM/Q   >
Pass /QSYS.LIB/ITSOIC00.LIB/HTML.FILE/*
#** End  :HTTP Directives for Team 00 ****************#
```

Figure 195. HTTP Configuration Directives for Team 00

Then, you need to change all of the places where you find ″00″ to ″01″ for team 01, ″02″ for team 02, and so on for all of your student teams.

3. Student PC requirements:

   The requirement for student PCs is not too demanding, but it does take some time to set up properly.

   - 486 or better based processor
   - We used OS/2 Warp, which allowed us to run both OS/2 WebExplorer (native OS/2 application) and Netscape (in a WIN-OS2 session) simultaneously. This allows students to compare and contrast the two clients side-by-side.
   - 16MB RAM
   - 15MB free hard disk
   - TRN or Ethernet connected to the AS/400 system
   - SVGA is nice - while VGA is OK.

     Note: Some display device drivers did a better job of color dithering for the graphics than others. More important than SVGA versus VGA is the number of colors. You should try to find a display and driver that supports 256 colors or more.

4. AS/400 system requirements:

   The requirements for the AS/400 system is also not too demanding. Most of the class runs quite well on the smaller models.

   - V3R2 of OS/400:
     - The GA CUM PTF package C6144320
     - The GA CUM+1 PTF package C6165320
     - A DB2WWW PTF SF34455 (5763-TC1)
     - A pair of PTFs for the HTTP server (SF32078 for 5763-TC1 and SF31077 for 5763-SS1) at a minimum.
     - PTFs SF33183, SF34734, and SF35356 should also be applied.

       Note: We have included a number of the PTFs that you might use in the library ITSOIC400 as save files. They are:

```
Object      Type        Attribute
SF31077     *FILE       SAVF
SF31879     *FILE       SAVF
SF32078     *FILE       SAVF
SF33183     *FILE       SAVF
SF34455     *FILE       SAVF
SF34734     *FILE       SAVF
SF35356     *FILE       SAVF
```

- Anonymous FTP requires PTF SF31879 for 5763-TC1 to be installed.
- TCP/IP Connectivity Utilities/400 (5763-TC1, a free LPP that comes with your purchase of OS/400) is loaded and configured.

  Note: The Internet Connection for AS/400 is a part of the TCP/IP Connectivity Utilities/400.
- 5763-PW1 Application Development ToolSet/400
- 5763-RG1 ILE RPG/400 - RPG/400
- 5763-WP1 OfficeVision/400

Appendix B. How Are OS/400 Users Counted for User-Based Pricing

User-based pricing was first introduced to the AS/400 system community back with the general availability of V3R1 of OS/400. The idea was that customers should pay for the number of users that use a particular system, not the size of the CPU. This appendix defines how OS/400 users are counted when using TCP/IP on different releases.

Release and CPU User-Based Pricing Rules

V3R1 on IMPI Every Telnet client that connects to your AS/400 system is counted as one OS/400 user. If two Telnet clients are coming from the same remote IP address (the same remote host), this is counted as two OS/400 users.

Note: When the Telnet client gets the Signon display, an OS/400 user is counted. So, beware of users that sit with the AS/400 Signon display for great lengths of time because this increases the count of OS/400 users. The reason for this strange behavior is that it is actually the AS/400 system′s Virtual Terminal API that is counting the OS/400 user, not the Telnet server. When the Telnet server requests a new virtual terminal, the API automatically adds one OS/400 user even though the next thing Telnet is going to do is simply put up the Signon display on the remote client.

All other TCP/IP applications such as FTP and user-written C-Sockets cause no OS/400 users to be counted.

V3R2 on IMPI The rules for the next IMPI release remain the same as V3R1. What is new in addition to the Telnet server is that each 5250-to-HTML Workstation Gateway client is also counted as one OS/400 user. So, both the Telnet server and the 5250 to HTML Workstation Gateway are counted as one OS/400 user for each client from Signon-to-Signon (inclusive).

V3R6 on PowerPC

For the RISC machines currently running V3R6 of OS/400, user-based pricing is counted at the sockets layer, not the Virtual Terminal API. Each unique remote IP address that comes into the AS/400 system is counted as one OS/400 user.

For example, let′s say that you use the FTP client on your DEC host to connect to the AS/400 FTP server. This counts as one OS/400 user. Then, you start transferring data that actually opens a second TCP connection between the DCE client and the AS/400 server, one for the control connection and the second for the data. Still, there is only one OS/400 user. Then, someone else on the same DEC host uses the Telnet client to sign on to the AS/400 system. Even with three TCP connections (and there could be many more) coming from the one DEC host IP address, only one OS/400 license is counted.

Note: Telnet sessions are not counted twice. In V3R6, the first Telnet session to come in to the AS/400 system from a remote IP address counts as one OS/400 user because it uses the socket layer. The code for counting at the Virtual Terminal API is not used in V3R6.

Appendix C. Special Notices

This publication is intended to help IBM Services Specialists in the delivery of Internet services pertaining to the AS/400 system. The information in this publication is not intended as the specification of any programming interfaces that are provided by the products mentioned in this book. See the PUBLICATIONS section of the IBM Programming Announcement for these products and 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, 500 Columbus Avenue, Thornwood, NY 10594 USA.

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 performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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:

AIX	AnyNet
Application System/400	APPN
AS/400	CallPath
DB2	DB2/400
IBM	OfficeVision
OfficeVision/400	OS/2
OS/400	PowerPC
PS/2	RS/6000
Ultimedia	WebExplorer
400	

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

AT&T	American Telephone and Telegraph
DEC VT52, DEC VT100 and DEC VT220	Digital Equipment Corporation
Network File System, NFS	Sun Microsystems, Inc.
Lotus, cc:Mail	Lotus Development Corporation
IPX	Novell, Inc.
Apple, Macintosh	Apple Computer, Inc.
X/Open	X/Open Company Limited
PKZIP, PKUNZIP	PKWARE, Inc.

Other trademarks are trademarks of their respective companies.

Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How To Get ITSO Redbooks” on page 301.

- *Using the Information Super Highway*, (GG24-2499)

D.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

D.3 Other Publications

These publications are also relevant as further information sources.

- *TCP/IP Configuration & Reference*, (SC41-3420)
- *TCP/IP Fastpath Setup Version 3*, (SC41-3430)
- *AS/400 Sockets Programming*, SC41-3422
- *AS/400 System API Reference*, SC41-3801
- *AS/400 Integrated File System Introduction (V3R1)*, SC41-3711
- *AS/400 Integrated File System Introduction (V3R6)*, SC41-4711
- *AS/400 NLS Planning Guide*, GC41-9877
- *ILE C/400 Programming Reference*, SC09-1677
- *ILE C/400 Programming Guide*, SC09-1520
- *ILE Guide*, SC09-1524
- *Tips and Tools for Securing Your AS/400* GC41-0615
- *IBM SecureWay AS/400 and the Internet* G325-6321

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.

- **REDBOOKS category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)** — ask for:

  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services** - send note to softwareshop@vnet.ibm.com
- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

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.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Index

Special Characters

Numerics

A

B

C

M

N

O

P

Q

R

S

T

U

V

W

Printed in U.S.A.