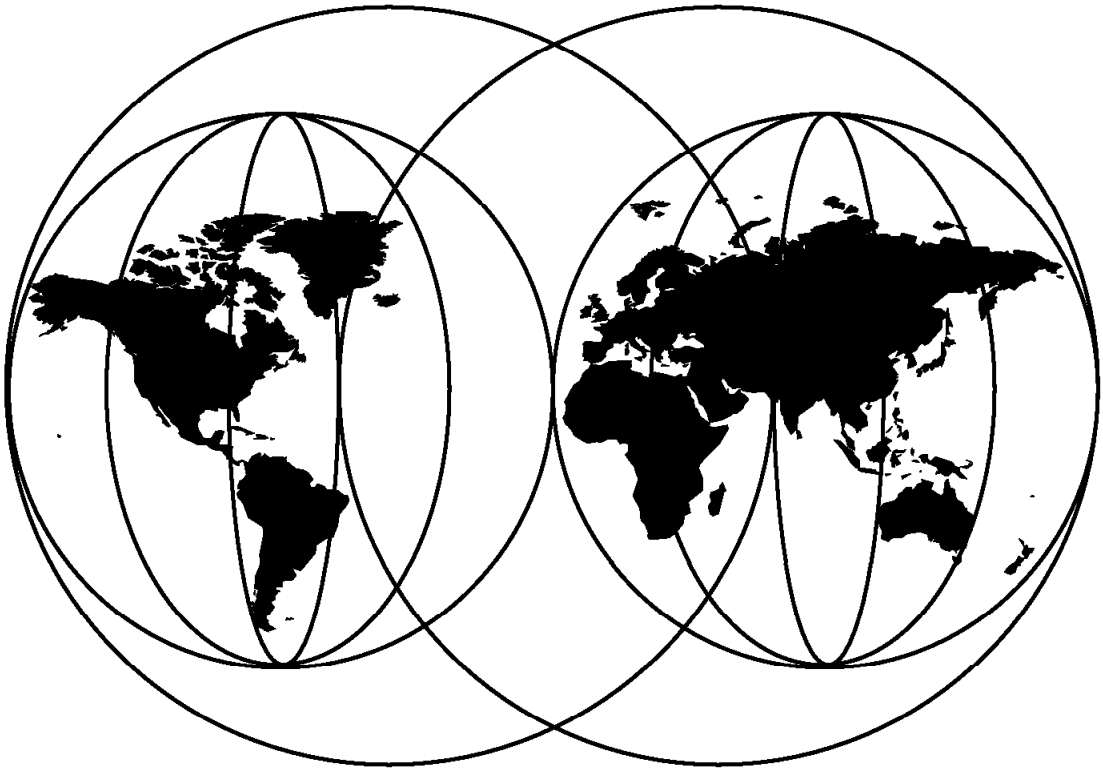
OS/390 Release 5 Implementation

Paul Rogers, Dr. Thomas Cornelius, Ralph Rudd, Andre Van Wyk



International Technical Support Organization

http://www.redbooks.ibm.com

SG24-5151-00

International Technical Support Organization

SG24-5151-00

OS/390 Release 5 Implementation

September 1998

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix D, “Special Notices” on page 247.

First Edition (September 1998)

This edition applies to Release 5 of OS/390, 5647-A01.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

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

Contents

Figures

Tables

Preface

This redbook describes the new functions available with OS/390 Release 5. These new functional enhancements are for the following components: OS/390 Print Server, ISPF from the World Wide Web, VisualAge for ISPF, Firewall Technologies on OS/390, Domino Go Webserver 4.6.1 for OS/390, SMP/E enhancements, HCD enhancements, and HCM enhancements.

This redbook will help you to tailor and configure the new functions of OS/390 Release 5. It can be used by systems programmers to install the new functions and understand the changes made to existing functions.

Chapter 1 contains an introduction and summary of all the functional enhancements available with OS/390 Release 5.

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 Poughkeepsie Center.

Paul Rogers	ITSO Poughkeepsie
Dr. Thomas Cornelius	Hypo-Bank, Munich Germany
Ralph Rudd	IBM South Africa
Andre Van Wyk	IBM South Africa

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

Paul de Graaff	ITSO Poughkeepsie
Rich Conway	ITSO Poughkeepsie
Roland Trauner	ITSO Poughkeepsie

Comments Welcome

Your comments are important to us!

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

- Fax the evaluation form found in “ITSO Redbook Evaluation” on page 263 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:

  For Internet users `http://www.redbooks.ibm.com/`
  For IBM Intranet users `http://w3.itso.ibm.com/`

- Send us a note at the following address:

  `redbook@us.ibm.com`

Chapter 1. OS/390 Version 2 Release 5 Overview

The OS/390 Version 2 Release 5 enhancements support OS/390 initiatives through support of critical base components for:

- Server consolidation
- Network computing
- Enterprise applications
- Business intelligence

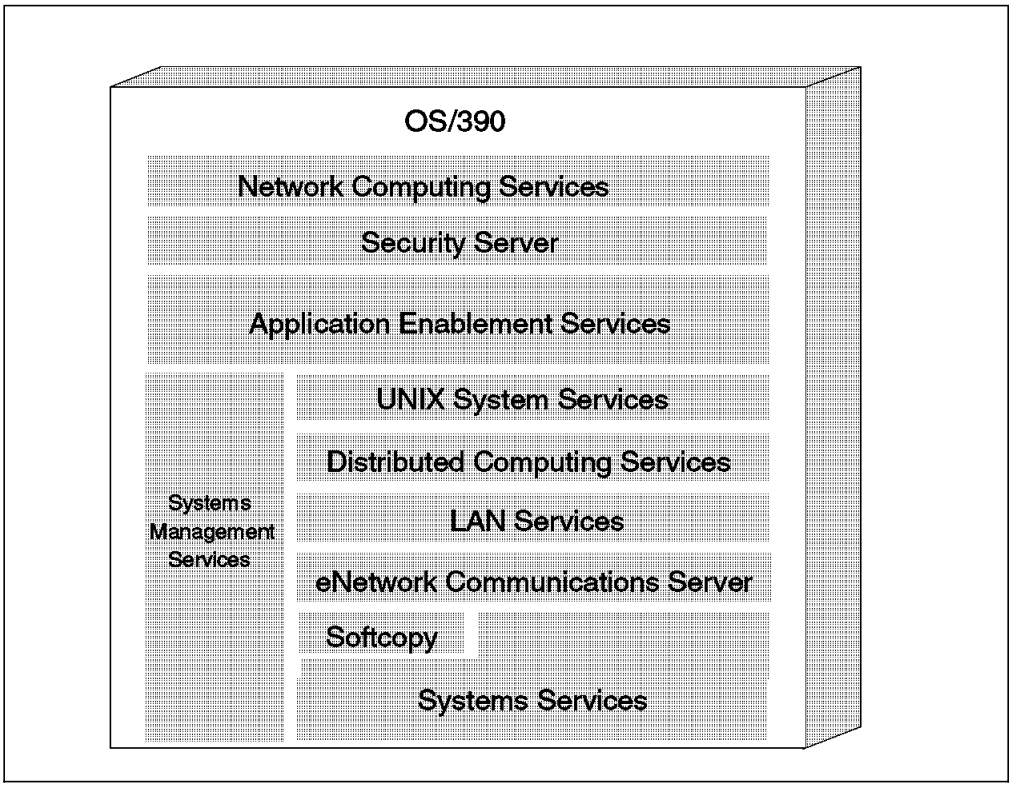


Figure 1. Overview of OS/390 Release 5 Structure

1.1 Server Consolidation Enhancements

Server Consolidation addresses the concerns of rising systems and operations management costs caused by the proliferation of multiple, mixed-vendor servers throughout many enterprises. OS/390 offers unique server consolidation function and capability in the following areas:

- A new OS/390 Print Server
- UNIX 95 (a UNIX brand function) and OS/390 UNIX System Services)
- Security Server enhancements
- An enhanced eNetwork Communications Server

1.1.1 OS/390 Print Server

Release 5 provides enhancements and new solutions enabling users to consolidate print workload from many servers in their IT environment onto a few S/390 servers. This can help reduce the overall cost of enterprise printing while improving usability, manageability and security.

Accessing printers in a network through a central print server is more cost effective than a distributed solution with printers attached to many different LAN servers, or even local desktop computers. Consolidating print onto a scalable server allows a user to use the right printer for specific print jobs, balance print workload across available printers, and achieve better management of an inventory of printers.

OS/390 Print Server, a new optional feature in Release 5, enables S/390 as an enterprise print server to handle host and LAN printing. The OS/390 Print Interface, a component of OS/390 Print Server, gives users of workstations, OS/2, AIX, Windows 95, and Windows NT the ability to print to OS/390-managed printers. The Print Interface also allows users and applications in the UNIX System Services environment to print on OS/390-managed printers. Printers can be high-speed AFP printers attached to OS/390 using PSF/MVS, or LAN-attached ASCII printers attached using IP PrintWay, which is a component of the OS/390 Print Server. OS/390 Print Server delivers improved efficiency and lower overall cost of printing, with the flexibility for high-volume, high-speed printing from anywhere in the network.

1.1.2 OS/390 UNIX System Services Enhancements

OS/390 UNIX System Services (formerly called OpenEdition) and the Bristol Wind/U product provide the flexibility to run UNIX applications and a set of Windows NT applications, respectively, on S/390. Release 5 delivers improved performance for OS/390 UNIX System Services. Individual functions have been restructured and rewritten to help applications reduce the number of address spaces resulting in both better performance and storage usage.

1.1.3 Security Server Enhancements

In Release 5, the OS/390 Security Server is enhanced to provide the Lightweight Directory Access Protocol (LDAP) Server that enables clients to add, delete, search, and extract information from an LDAP server that resides on an S/390 platform.

1.1.4 eNetwork Communications Server

For OS/390 Release 5, the eNetwork Communications Server provides a set of communications protocols that support peer-to-peer connectivity functions for both local and wide-area networks, including the most popular wide-area network, the Internet. eNetwork Communications Server also provides performance enhancements such as high throughput for file transfers.

Key new facilities in eNetwork Communications Server are:

- A new, completely revamped TCP/IP service improves performance and reliability for TCP/IP users.
- A new TN3270E server offers improved performance and 3270 services.
- TCP/IP (DNS) workload balancing for Parallel Sysplex users and automated registration of clients to DNS is supported.

- Improved Internet and intranet security, special support for “very high demand” OS/390 Web serving environments, native ATM high speed networking for IP networks, and other significant improvements are offered in this release.
- Integration of Firewall Technologies into the OS/390 Security Server and eNetwork Communications Server.

1.2 Network Computing Enhancements

The following items are new or enhanced in OS/390 Release 5 for the Network Computing initiative:

- Domino Go Webserver for OS/390 4.6.1
- Support for Java Development Kit (JDK) 1.1
- Digital Certificates
- ICSF: Finance and Commerce enhancements
  - SET protocol support for safeguarding payment card purchases made over open networks.
  - Triple DES support for data privacy (software that exploits this function cannot be exported outside North America without a special license from the United States government).
  - Support for card verification code (CVC) and card verification values (CVV) reduces the risk of losses resulting from alteration/counterfeiting and provides additional integrity for financial processes involving payment cards.

1.2.1 Lotus Domino Go Webserver 4.6.1

The Domino Go Webserver 4.6.1 for OS/390 has the following enhancements:

- %%CERTIF%% Support: This support allows the OS/390 Security Server to receive a certificate from the Domino Go Webserver and use the certificate in place of a user ID and password.
- JAVA Servlet Support: JAVA Servlet external process support is enabled under OS/390 based on the latest available JDK 1.1 driver.
- Performance Enhancements: Improvements have been made to the processing of base server requests, yielding an overall increase in request throughput.
- Proxy Enhancements: Numerous changes have been implemented and tested based on customer environments. These changes have also been field tested and resulted in overall RAS improvements to the Webserver.
- Security: Upgrading certificate roots and the request process (certutil) is enhanced or replaced to issue online and offline certificates to Domino Go Webserver servers, other vendor servers, and client certificates for Netscape browsers. The user is able to import certutil-generated CA roots into the Netscape and MSIE browsers. Support for other vendor CA roots is available, in addition to Verisign.

1.3 Enterprise Applications

The S/390 Applications Initiative′s goal is to provide new applications, new infrastructure in support of programming environments, and new support for application growth through tools for S/390 and OS/390. The Application Initiative focus is based on:

- Object-Oriented Component Broker technology
- OS/390 UNIX System Services technology
- Technologies that optimize application development, porting and execution

1.3.1 Component Broker

Component Broker for OS/390 provides the ability to develop and deploy mission-critical Object-Oriented (OO) technology applications that leverage the traditional strengths of the S/390 platform. This is important in reducing the traditional costs associated with application development and reducing turn-around time.

1.3.2 Application Enabling Technology

OS/390 Application Enabling Technology (OS/390 AET) Enhancements was first announced and available in March, 1997. In Release 5, it is enhanced to provide Lotus Domino for S/390 and network computing support as well as enhanced system administration and control center capabilities. Users can easily customize and create standalone or distributed Lotus Domino for S/390, Domino Go Webserver, and Network Station solutions for their groupware and Internet/intranet requirements.

1.4 Additional OS/390 Release 5 Enhancements

OS/390 Release 5 also includes some enhancements to ISPF and SystemPac.

1.4.1 ISPF Enhancements

Two new components have been added to the ISPF product for OS/390 Release 5:

- ISPF web access enhancements

  This component includes an ISPF Application Server and ISPF Workstation Agent Applet, which allow legacy and new ISPF application to be accessed from the World Wide Web. This allows ISPF applications to be run from a network computer or workstation browser utilizing Java 1.1.

- VisualAge ISPF

  This component is a visual development solution utilizing the composition editor of VisualAge technology, which allows the creation of new ISPF panels and modification of existing ISPF panels for use on 3270 and GUI screens. This gives users the flexibility to be able to create or modify ISPF panels without needing to know the syntax of the ISPF panel language or ISPF Dialog Tag Language (DTL).

1.4.2 SystemPac Enhancements

SystemPac offers the capability of building a system with integrated subsystems in both copy format and full volume dump/restore format. Other than IBM products, Independent Software Vendor products can be selected and included with the SystemPac. After the delivery of the SystemPac, Selective Follow On Service tapes (Hipers and PTFs resolving PEs) can be shipped at specified intervals and frequency based on customer's selection at ordering time.

When ordering the OS/390 Release 5 SystemPac, you are now able to specify the following additional customization items using the local order entry system in addition to customization gathered previously. These new customization capabilities are:

- For full volume dump/restore format only:
  - Specification of SYSNAME
  - GLOBAL CSI name along with the capability of zone assignments to the respective target/dlib CSIs
  - Setup of indirect referencing
- For both full volume dump/restore format and copy format:
  - Capability to display PRODUCT NAMEs/PGMIDs besides high-level qualifiers of selected products, which eases the task of assigning aliases to specified catalogs
  - Specification of the master catalog name
  - Capability to change data set names up front and have the system built according to the specified naming conventions
  - Capability to input specific variables (for example, SVC for IMS) and have products set up according to these variables
  - The local order entry panels used for providing the package customization input now contain Help information for every panel.

1.5 Release 5 Enhancements for Data Consolidation

To provide effective and efficient support for data consolidation, enhancements are available in Release 5 as follows:

1.5.1 LAN Services

OS/390 LAN Resouce Extensions and Services (LANRES) also delivers currency for NetWare distribution and print support. LANServer and NFS, although unchanged in this release, continue to provide LAN data consolidation options. The four elements (LANRES, LANServer, NFS, and DFS) deliver the ability to place the LAN data on S/390 and take advantage of S/390's data management, integrity and backup, while reducing the number of middle tier servers.

1.5.2 Distributed Computing Environment

Distributed File Services (DFS) delivers performance, Open Systems Foundation (OSF), message, installation/configuration, and backup support enhancements.

1.5.3 SMP/E Enhancements

This release of the SMP/E element of OS/390 focuses on performance, usability, and application growth capability as follows:

- Parallel multiple link-editor operations

  This performance enhancement enables multiple link-edit operations to occur in parallel when the link-edit utility is reentrant and certain utility files can be dynamically allocated based on previous allocations. This parallelism shortens elapsed time of an SMP/E APPLY, ACCEPT, or RESTORE when a large number of SYSMODs are being processed and several libraries are being updated by the link-edit utility.

- Client code installation

  A common SMP/E packaging structure, a common S/390 server repository for client components, and a server repository that is accessible from any client platform are provided. This capability enables facilities to make the installation of cooperative or client/server products more seamless from the user's perspective.

- Improved load module build processing

  This function does not allow SMP/E to build a load module without including all of its component modules that have been installed or are being installed. It performs an expanded search for these component modules. This capability reduces the likelihood of SMP/E incorrectly building a new load module during APPLY processing. More importantly, it reduces the likelihood of termination because of incomplete load modules.

- Global zone merge

  This capability provides a method for merging information from one Global zone into another Global zone. This function is particularly useful to ServerPac customers.

- SMPPTS data set compaction

  To reduce the space requirements of the SMPPTS data set, SMP/E compacts PTF members within the data set during RECEIVE processing and expands them during APPLY and ACCEPT processing. SMP/E also provides a standalone service routine, GIMCPTS, which can be used to compact or expand PTFs outside of the RECEIVE, APPLY, and ACCEPT context. DASD space needs are reduced during PTF installation.

  The Report ERRSYSMODS command is updated to use the new enhanced HOLDDATA that is provided in the ++HOLD statement.

- Enhanced RECEIVE command

  This capability enables SMP/E to optionally not RECEIVE SYSMODs that are already APPLIED and/or ACCEPTED. This provides relief because the customer does not have to manually manage the SMPPTS using REJECT processing.

- Reduce and simplified SMP/E messages

  This function provides for quicker identification of potential problems by reducing the number of messages issued during APPLY, ACCEPT, and RESTORE processing.

- Library Change Interface

This capability provides a general-use programming interface that contains a synopsis of the processing done via SMP/E APPLY/RESTORE at the library/member level. This information serves as input to a multisystem software distribution application/process (cloning/propagation).

- Support for all Entries and Subentries in API

  This capability allows the user to retrieve the Consolidated Software Inventory (CSI) data for all entry types and/or all subentries.

- API version support

  This capability supplies the user with the version of the API that is being executed to retrieve information from the CSI.

- Load module return code

  This function provides additional granularity for the highest acceptable return code values that are used for all link edit operations during SMP/E command processing.

- S/390 update facility

  OS/390 Release 5 SMP/E, along with other S/390 products, provides a common tool across multiple platforms to help customers maintain their systems with S/390 service facilities.

1.5.4 OS/390 Hardware Configuration Dialog

OS/390 Release 5 is enhanced to provide a better way to configure and manage large amounts of data and is changed as follows:

- HCD verification and priming of I/O configuration (Stage II)

  This is the second stage of support introduced in OS/390 Version 1 Release 3. This stage supports the priming of device self-description data, such as serial numbers and ESCON director port connections, not only as a separate step but also during the definition of new configuration elements. Furthermore, it supports creating and updating the CONFIGxx member from the IODF definition. This function requires that System Automation for OS/390 (SA OS/390), the follow-on product for ESCON Manager, be installed and active.

- HCD large IODF and distributed configuration enhancements

  This function enables the distribution of single configurations out of an IODF to a target system and also the merging of distributed IODFs to a master IODF.

  This function includes keeping the processor token in sync within a sysplex, and provision of distribution lists. This function provides relief for the primary user address space by accessing the IODF data via a separate data space.

1.5.5 OS/390 Hardware Configuration Management

The changes for hardware configuration management (HCM) are as follows:

- HCM Windows 95 and Windows NT client support

  The Windows 95 and Windows NT client support enables HCM to run on Windows 95 as well as on Windows NT platforms. This support is also available via SPE PTFs for HCM V.1.1.0 and OS/390 Version 2 Release 4

HCM. This support eliminates the current restriction of HCM to only run on Windows 3.x and Win-OS/2.

- HCM enhanced filter capabilities for HCM diagrams

  This feature allows a configuration to be examined and searched by using a sophisticated, flexible, SQL-like data viewing mechanism. Furthermore, HCM provides the ability to save and maintain a set of predefined filters for easy recall. This support is also available via SPE PTFs for OS/390 Version 2 Release 4 HCM. This makes it significantly easier to manage large amounts of configuration data.

Chapter 2. OS/390 Print Server

With the introduction of the OS/390 Print Server, a new optional feature of OS/390 Release 5, users have the opportunity to consolidate their print workload on OS/390. This new function allows access to fast and reliable AFP printers, or TCP/IP-connected printers from OS/390, including UNIX services and LAN clients. Users can define their printers in a central repository, allowing clients in the network to use any printer in the enterprise that is registered to the OS/390 Print Server.

The OS/390 Print Server is the framework for a total print serving solution for the OS/390 system environment. It extends the functions provided by the optional IP PrintWay and NetSpool features in OS/390 Version 1 Release 3 and OS/390 Version 2 Release 4. IP PrintWay and NetSpool are now part of the OS/390 Print Server.

With the addition of a new Print Interface component, the OS/390 Print Server provides an end-to-end integrated solution from print submission to the printer. The IBM OS/390 Print Server consists of three components, as shown in Figure 3 on page 11, as follows:

- OS/390 Print Interface

  The OS/390 Print Interface is the central server element of the IBM OS/390 Print Server. It consists of an LPD that allocates data sets on the JES spool using information from the printer definitions in the printer inventory. For more information on the Print Interface, see 2.2, “OS/390 Print Interface” on page 10.

- IP PrintWay

  IP PrintWay can use standard LPR/LPD or direct socket printing protocol to route JES2 or JES3 print data from OS/390 to another system′s spool or to a printer in the TCP/IP network. Depending upon selected options, the print data is sent as is (binary format), or translated from EBCDIC into ASCII for the target system or printer. IP PrintWay is better than the TCP/IP Network Print Facility (NPF) for MVS in usability, performance, capacity and function, and is the strategic replacement for NPF.

- NetSpool

  NetSpool allows the user to automatically reroute VTAM application output (such as from CICS or IMS) to the JES spool without requiring application program changes. Application output can then be printed to any server or printer that is connected to the TCP/IP network using IP PrintWay, or to an AFP printer using Print Services Facility (PSF)/MVS.

  Note: This component of the IBM OS/390 Print Server is optional.

2.1 TCP/IP Print Protocol

TCP/IP provides client and server support for remote printing by supporting the following commands:

LPR Line print requestor (LPR) is the command that requests printing. The standard LPR command has attributes such as file name, IP address, and queue name. The LPR command sends print data to an LPD.

LPD Line print daemon (LPD) is like a destination. An LPD daemon waits for the LPR command to be invoked. The LPD daemon receives the print data and sends it to a print queue. It can reside in the software or in the printer hardware, as shown in Figure 2 on page 10.

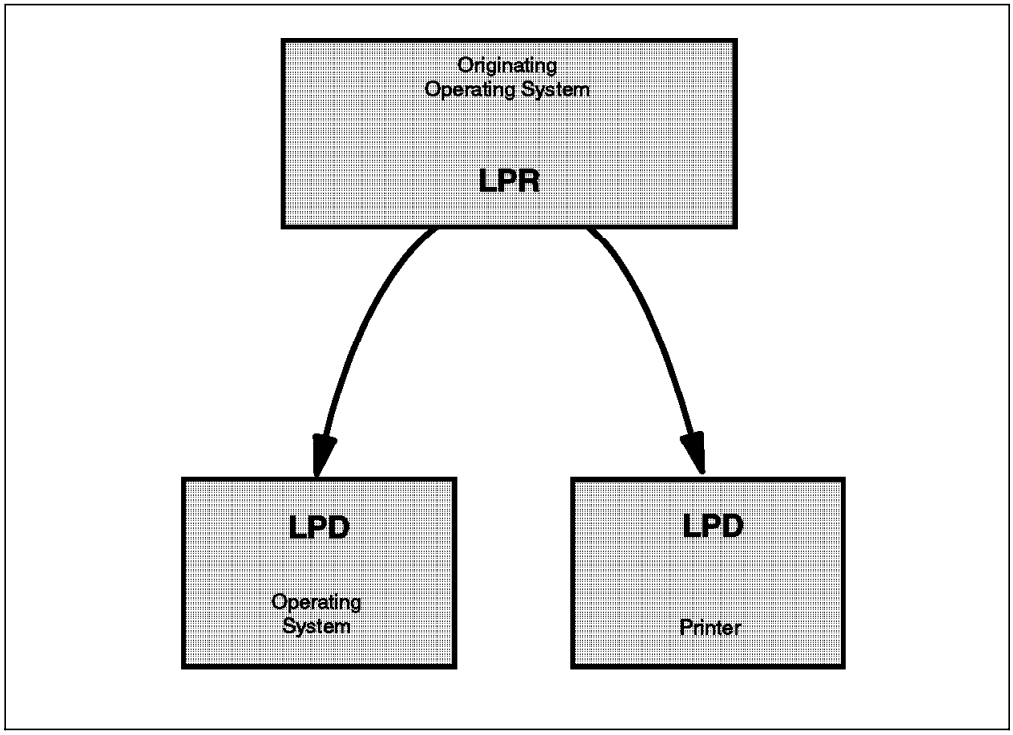


Figure 2. TCP/IP Printing

The LPR command and the LPD function are part of IBM TCP/IP Version 3 Release 2 of the OS/390 base feature.

You can print from the following environments by using the TCP/IP LPR command:

- AIX 4.1.4 or 4.2.x
- OS/2
- VM/CMS
- A remote OS/390 system using TSO/E
- A local OS/390 system using TSO/E

2.2 OS/390 Print Interface

The OS/390 Print Interface, shown in Figure 3 on page 11, runs as an LPD on the OS/390 system. It provides the following functions:

- It receives print requests and dynamically allocates a data set on the JES spool for each data set to be printed.
- It responds to query requests and returns the status of the data set on the JES spool or a list of printer names, locations, and descriptions.
- It removes data sets from the JES spool that have not been selected for printing.

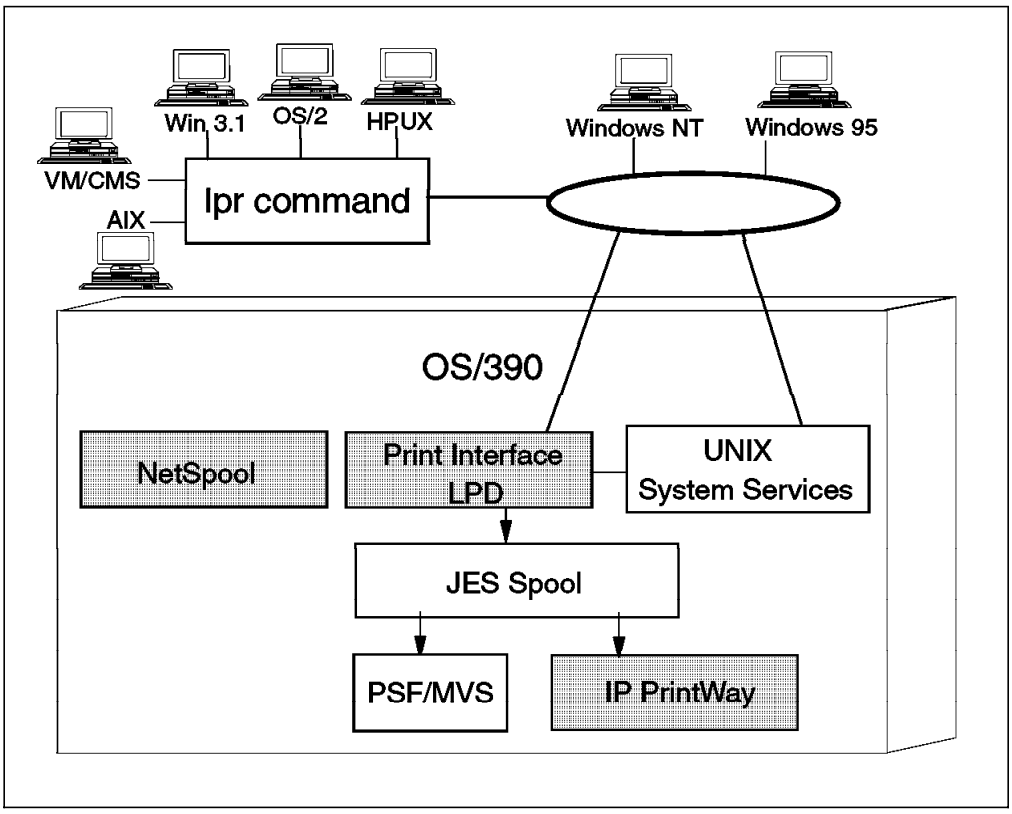


Figure 3. OS/390 Print Server Printing

The OS/390 Print Interface, shown in Figure 3, receives print requests that are submitted using TCP/IP protocol from:

- Remote systems in the TCP/IP LAN network
- OS/390 UNIX System Services (OpenEdition)
- A Windows 95 or Windows NT system
- A local OS/390 system

2.2.1 Remote Systems in the TCP/IP LAN Network

The following operating systems, shown in Figure 3, can use the LPR command to send print requests to the Print Interface:

- AIX 4.1.4 or 4.2.X
- OS/2
- VM/CMS
- HPUX
- Win 3.1

The job submitters can use the following TCP/IP commands:

lpr To print documents

lpq To query the status of printing

lprm To cancel print requests

2.2.2 OS/390 UNIX System Services (OpenEdition)

When printing from UNIX System Services, you can print the following types of data:

- Hierarchical File System (HFS) files
- Partitioned data sets
- Sequential data sets

A job submitter can use the following commands:

lp Print documents

lpstat Query the status of print requests

cancel Cancel print requests

See 2.7, “Printing from UNIX System Services” on page 34 for the details.

2.2.3 A Windows 95 or Windows NT System

A job submitter can print documents from any Windows application using:

- Standard methods of print submission available with Windows applications
- A Windows (NT or 95) client provided with an IBM OS/390 Print Server (see 2.12.1, “Printing from Windows 95 and Windows NT” on page 41)

The job submitter′s destination printer can be defined in the OS/390 Print Interface printer inventory.

The Windows client passes job and document attributes to the OS/390 Print Interface. The job name, owner, and the requested printer name are passed to the server.

Note: The OS/390 Print Interface does not return error messages or other job process notifications to these clients. In addition, Windows clients cannot query the status of print requests or cancel a print request.

Note: The OS/390 Print Server Port Monitor must be installed.

2.3 OS/390 Print Interface Customization

OS/390 UNIX Services must be installed to use the OS/390 Print Server.

Following your system install, a configuration file called *aopd.conf* contains configuration information to be used by the OS/390 Print Interface and by the OS/390 UNIX System Services printing commands provided with the OS/390 Print Server.

The configuration file is stored in an HFS. It is shown in Figure 4 on page 13.

```
# default aopd configuration
server-port = 515
base-directory = /var/Printsrv
oids-directory = /usr/lpp/Printsrv/oids
ascii-codepage = ISO8859-1
ebcdic-codepage = IBM-1047
job-prefix = PS
```

Figure 4. OS/390 Print Interface Configuration File

Copy the sample configuration file from:

/usr/lpp/Printsrv/samples/aopd.conf

To:

/etc/Printsrv/aopd.conf

By using the ISHELL TSO/E command to perform the copy and customization as follows:

- From Option 6 in ISPF invoke the ISHELL, as shown in Figure 5.

```
 Menu  List  Mode  Functions  Utilities  Help
------------------------------------------------------------------------
                              ISPF Command Shell
Enter TSO or Workstation commands below:

===> ishell



Place cursor on choice and press enter to Retrieve command

=>
=>
=>
=>
=>
=>
=>
=>
=>
=>
```

Figure 5. From Option 6 in ISPF, Enter ISHELL

- Type in the /samples directory, as shown in Figure 6 on page 14, and press Enter.

```
  File  Directory  Special_file  Tools  File_systems  Options  Setup  Help
 ------------------------------------------------------------------------
                            OpenMVS ISPF Shell

 Enter a pathname and do one of these:

     - Press Enter.
     - Select an action bar choice.
     - Specify an action code or command on the command line.

 Return to this panel to work with a different pathname.
                                                                  More: +
    /usr/lpp/Printsrv/samples
    ____________________________________________________________________
    ____________________________________________________________________
    ____________________________________________________________________


 Command ===> ____________________________________________________________
```

Figure 6. Enter the /samples Directory

- A list of the files that exist in the /samples directory is displayed. Type in the c (copy) action code next to the init.options file and press Enter.

```
                                Directory List

 /usr/lpp/Printsrv/samples/
 Select one or more files with / or action codes.

   Type  Filename                                              Row 1 of 6
 _ Dir   .
 _ Dir   ..
 C File  aopd.conf
 _ File  cfilter.c
 _ File  cfilter.h
 _ Dir   IBM
```

Figure 7. Select the File to Be Copied

- The next panel shows the from file and you should select Option 1 (copy is to another file) and press Enter.

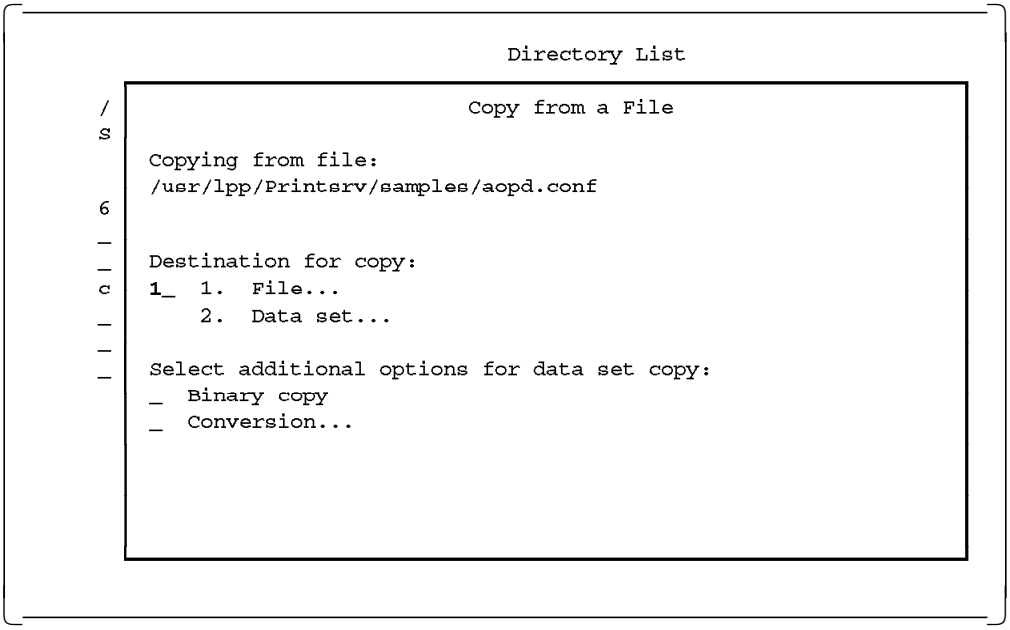


Figure 8. Panel to Select Destination Type

- The next panel shows the “from” file, **/usr/lpp/Printsrv/samples/aopd.conf**, and you should overtype this with the “to” file and press Enter.

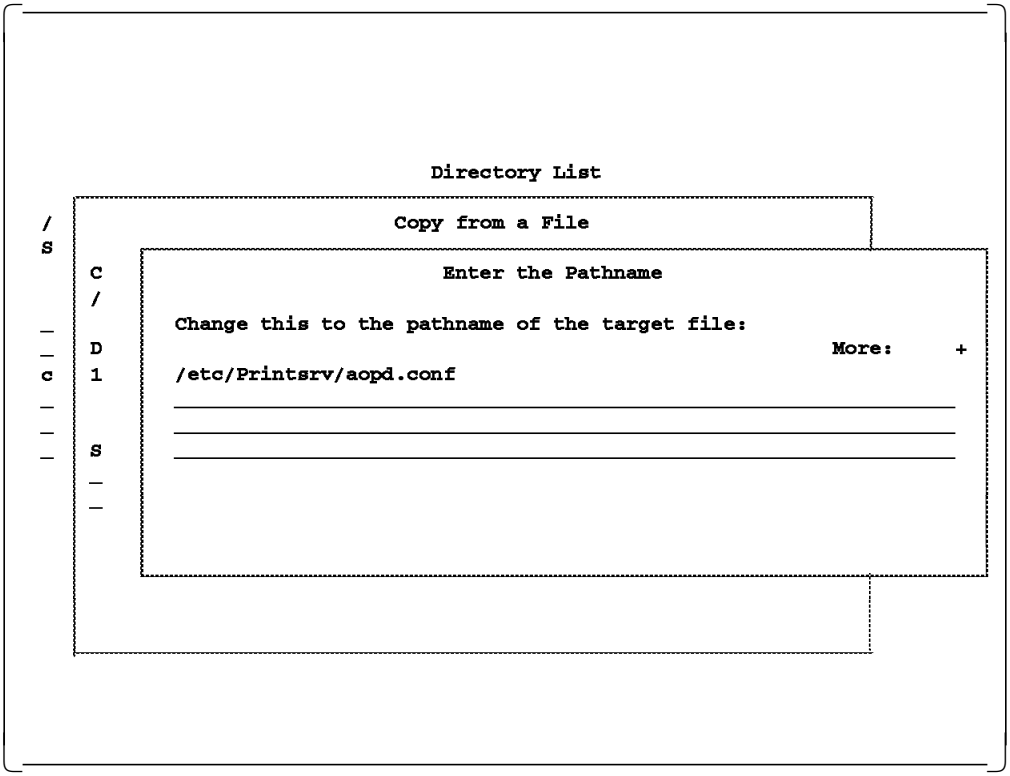


Figure 9. Panel to Select the Directory for the Copy

- The next panel in Figure 10 on page 16 lets you determine the permission bits to be set.

You control access to a file and directory that you own through its permission bits. The permission bits are often called the *mode*

When you first create a file or directory, the system sets default read, write, and execute (rwx) permissions. The meanings of the three permissions for a file are:

READ (r) Permission to read or print contents. To run a shell script, you need both read and execute permission.

WRITE (w) Permission to change the file, adding or deleting data.

EXECUTE (x) Permission to run a file, that is, enter it as a command. Typically this permission is used for shell scripts and for files containing executable programs. To run a shell script, you need read and execute permission.

Note: For a description of permission bit settings, see Appendix B, “Permission Bits” on page 197.

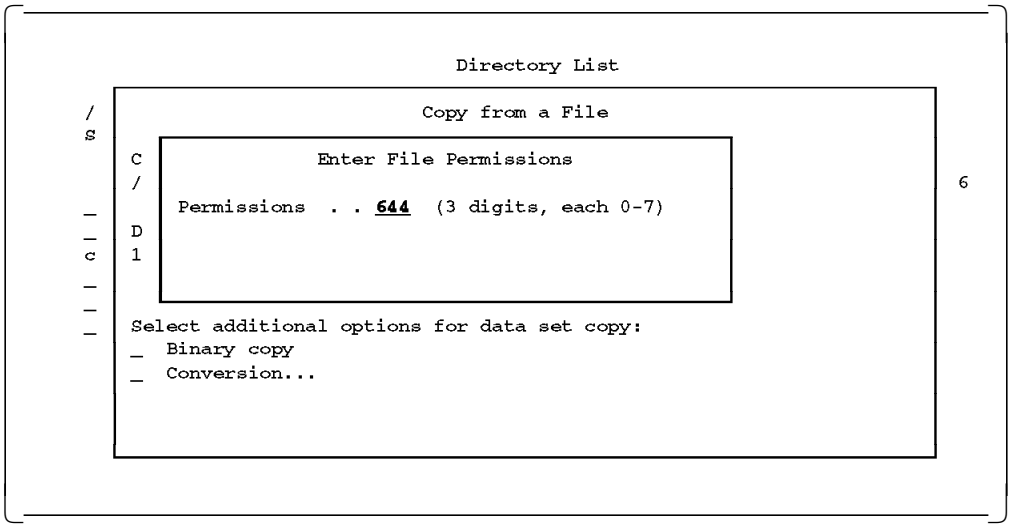


Figure 10. Panel to Select Permission Bits

2.3.1 Print Interface Authorization

The Print Interface load modules are stored in AOP.SAOPLOAD. This data set must be APF authorized. See *Program Directory for OS/390 Version 2 Release 5 For CBPDO Installation and ServerPac Reference*.

2.4 Defining Printers for Print Interface

The Print Server provides an ISPF interface for defining printers to the Print Interface. The primary option panel is shown in Figure 11 on page 17.

You could create one printer definition for each local printer in your OS/390 system and one printer definition for each remote print queue in your TCP/IP network. You might want to consider other configurations of printer definitions for PSF/MVS or General printer definitions.

```
------------ OS/390 Print Interface: Printer Definition Management -----------
COMMAND ===>                                                 USERID - ROGERS
                                                             DATE   - 98/03/27
Type an option on the command line and press Enter.          TIME   - 17:29

 1 Select      - Select printer definitions to list.  From the list,
                 you can edit, copy, or delete definitions.

 2 List        - List all printer definitions.  From the list,
                 you can edit, copy, or delete definitions.

 3 Add         - Add printer definitions.


 4 List models - List models for printer definitions.
                 Models allow you to change defaults for printer definitions.

 5 Configure   - Configure panel options and set default printer.


 6 Exit        - Exit OS/390 Print Interface panels
```

Figure 11. Print Interface Primary Option Panel

Option 3 on the panel shown in Figure 11 lets you add printers to the Print Interface. There are three types of printers, as shown in Figure 12:

- IP PrintWay printers to be controlled by IP PrintWay
- PSF/MVS printers to be controlled by PSF
- General printers

```
------------ OS/390 Print Interface: Printer Definition Management -----------
             OS/390 Print Interface: Add New Printer Definition
------------------ OS/390 Print Interface: Add a Printer ------------------
COMMAND ===> ____________________________________________    USERID - ROGERS
                                                            DATE   - 98/03/27
     Select the type of printer definition to add:          TIME   - 18:11

     _ 1. IP PrintWay

       2. PSF/MVS

       3. General

          Press ENTER to access the Add Printer Definition panel.


 5 Configure   - Configure panel options and set default printer.


 6 Exit        - Exit OS/390 Print Interface panels
```

Figure 12. Option 3 to Add Printers to Print Interface

Printers can be added dynamically without any interruption of the Print Interface or any of the workstations active with the Print Interface.

2.4.1 Defining IP PrintWay Printers

Figure 13 and Figure 14 on page 19 show the panels to create printers that are to be controlled by IP PrintWay.

Note: Supplied with the Print Server product is a utility program that converts existing IP PrintWay routing entries to Print Interface defined printers. See 2.4.1.2, “Migration Utility for IP PrintWay Printers” on page 19.

Note: If you are using an IP PrintWay routing data set, do not specify a Printer IP Address, Print Queue Name, Options Entry, Retry Time, or Retain Time shown in Figure 13.

The Destination Node may be specified to send output to another node.

```
ADD    -------- OS/390 Print Interface: IP PrintWay Printer Definition -------
COMMAND ===>                                                 USERID - ROGERS
                                                             DATE   - 98/03/27
  Printer Name  ===> _____________                           TIME   - 18:40
                                                                    More:     +
  Description ===> ____________________________________________________
  ____________________________________________________________________

  Printer Location ===> ___________________________________________
  ____________________________________________________________________

 Work Selection or Routing Selection:
   CLASS     ===> _    DEST         ===> ________   FORMS  ===> ________
   Priority  ===> ___  Process mode ===> ________   Writer ===> ________

 Transmission Attributes:
   Print Queue Name ===> ____________________________________________
   ____________________________________________________________________

   Printer IP Address ===> ____________________________________________
           ____________________________________________________________

   Options Entry ===> _________________

   Retry       : Time                (HHHH:MM:SS)
               : Limit   __________   (0-32767)
   Retain Time : Success __________   (HHHH:MM:SS or FOREVER)
               : Failure __________   (HHHH:MM:SS or FOREVER)

   Destination Node ===> ________

 Code Pages: Document ===> _______________    Printer ===> ISO8859-1___
```

Figure 13. Defining IP PrintWay Printers to the Print Interface (Screen 1 of 2)

```
-------------- OS/390 Print Interface: Validation Attributes --------------

Maximum Job Size:
 1  1. No Limit
    2. Bytes  ===> ___________(1-2147483647)

Maximum Copies:
 1  1. No Limit
    2. Number ===> _____       (1-255)

Data Formats Supported and associated Filter Paths:
                                                                 ("/" or blank)
  Data Format                      Filter Path
 ============= ============================================================
 / Line-data   ____________________________________________________________
 / MO:DCA-P    ____________________________________________________________
 / PostScript  ____________________________________________________________
 / Text        aopfiltr.so________________________________________________
 / PCL         ____________________________________________________________
 / Other       ____________________________________________________________
```

Figure 14. Defining IP PrintWay Printers to the Print Interface (Screen 2 of 2)

2.4.1.1 Filter Programs

Figure 14 shows a default filter program for text data with IP PrintWay printers. IBM provides this filter program, named aopfiltr.so, to convert line-feed controls (X'0A') that are not preceded by carriage-return controls to carriage-return and line-feed controls (X'0D0A'). The X'0D0A' control is suitable for most ASCII printers and print queues. The aopfiltr.so program is installed in directory /usr/lpp/Printsrv/lib. This filter program is intended for ASCII data and is only a default for IP PrintWay printers in the ISPF panels.

IBM recommends that you specify the aopfiltr.so program for ASCII printers when the data format is text.

Note: For information on writing filter programs, see *OS/390 Print Interface Configuration Guide*, G544-5544.

2.4.1.2 Migration Utility for IP PrintWay Printers

If you are currently running IP PrintWay and have defined routing entries in the IP PrintWay routing data set, you can convert this data set to Print Interface printer definitions.

Figure 15 on page 20 shows the migration program JCL to convert IP PrintWay printer definitions from the routing file into OS/390 Print Interface definitions. This sample JCL is in SYS1.SAMPLIB, member aopmigj.

Note: See *OS/390 Print Interface Configuration Guide*, G544-5544 for details of this migration program.

```
//AOPMIGJ  JOB MSGLEVEL=(1,1)
//*************************************************************
//*
//*     OS/390 Print Interface - FMID HOPI100
//*
//* This sample JCL will run the migration program to convert
//* IP PrintWay printer definitions for OS/390 Print Interface
//*
//* Make the following modifications before running this job:
//* 1) Change the job card to meet your system requirements
//* 2) Change ANF.ROUTING on the AOPRVSAM DD statement to point
//*    to your IP PrintWay routing file
//*
//*************************************************************
//TSOBATCH EXEC PGM=IKJEFT01,DYNAMNBR=25,REGION=2M
//SYSPROC   DD  DSN=SYS1.SAMPLIB,DISP=SHR
//SYSPRINT  DD  SYSOUT=*
//SYSTSPRT  DD  SYSOUT=*
//SYSTSIN   DD  *
 aopmigr
 ocopy indd(STDOUT) outdd(SYSTSPRT) pathopts(override)
 ocopy indd(STDERR) outdd(SYSTSPRT) pathopts(override)
/*
//AOPRVSAM  DD  DSN=ANF.ROUTING,DISP=SHR
//AOPRFLAT  DD  UNIT=SYSDA,SPACE=(CYL,1),
//          DCB=(LRECL=640,BLKSIZE=8192,RECFM=VB)
//STDOUT    DD  PATH='/tmp/migrat.stdout',
//            PATHMODE=SIRWXU,PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//STDERR    DD  PATH='/tmp/migrat.stderr',
//            PATHMODE=SIRWXU,PATHOPTS=(OWRONLY,OCREAT,OTRUNC)
//STDENV    DD   *
AOPCONF=/etc/Printsrv/aopd.conf
PD_CONFIRM_DELETE=no
/*
```

Figure 15. Migration JCL to Convert IP PrintWay Routing File

2.4.2 Defining PSF/MVS Printers

Figure 16 on page 21 and Figure 17 on page 22 show the panel for defining a PSF/MVS printer to the Print Interface. You should create a PSF/MVS printer definition if the printer is controlled by PSF/MVS.

In a PSF/MVS printer definition, you can specify attributes that PSF/MVS uses to print documents, such as the name of a form definition and the name of a page definition.

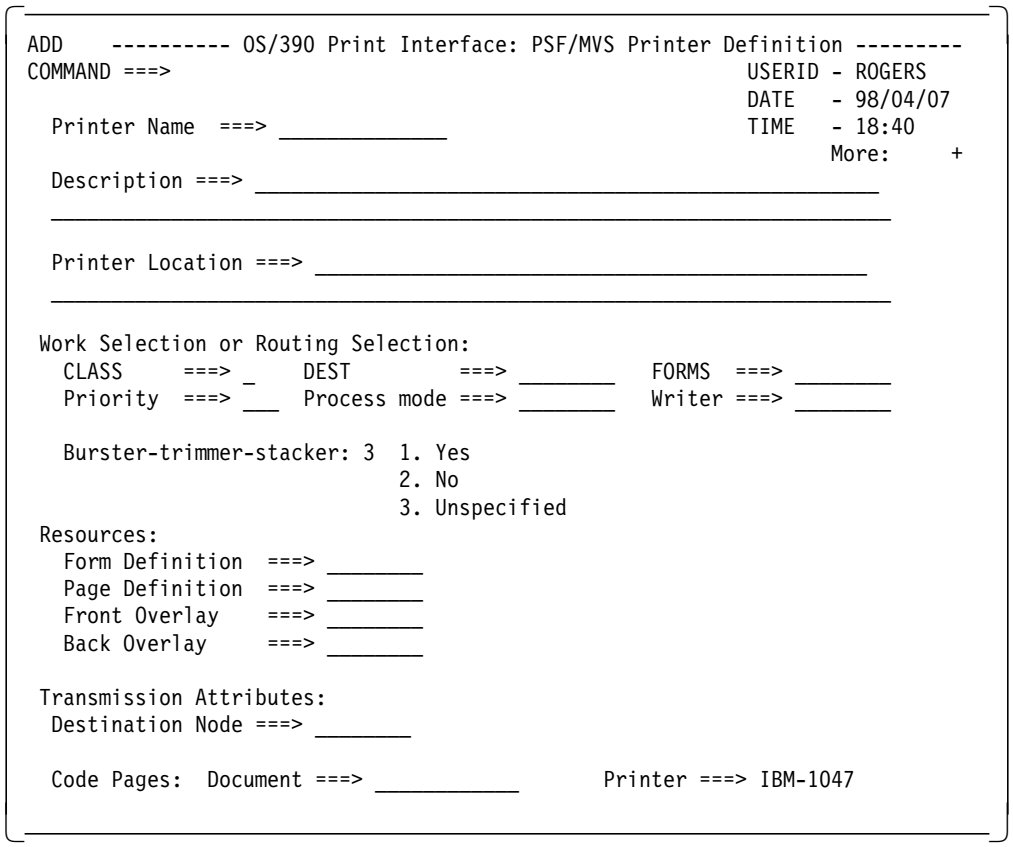


Figure 16. Defining PSF/MVS Printers to the Print Interface (Screen 1 of 3)

```
---------- OS/390 Print Interface: Validation Attributes ----------

Maximum Job Size:
 1  1. No Limit
    2. Bytes  ===> __________  (1-2147483647)

Maximum Copies:
 1  1. No Limit
    2. Number ===> ___         (1-255)

Forms Supported:
 1  1. All Forms
    2. Form ===> ________  ________  ________  ________  ________

Data Formats Supported and Associated Filter Paths:
                                                          ("/" or blank)
  Data Format                       Filter Path
 ============= ========================================================
/ Line-data    ____________________________________________________
/ MO:DCA-P     ____________________________________________________
  PostScript   ____________________________________________________
/ Text         ____________________________________________________
  PCL          ____________________________________________________
  Other        ____________________________________________________

Duplex Supported:  / Simplex    / Duplex     / Tumble       ("/" or blank)

Print-error Reporting Supported:  / Character  / Position   ("/" or blank)
```

Figure 17. Defining PSF/MVS Printers to the Print Interface (Screen 2 of 3)

```
------ OS/390 Print Interface: Input Tray and Output Bin Attributes ------

Input Trays:                          Output Bins:

      Name          Number                  Name          Number
      Alternate       1                     Bottom          _____
      Bottom          2                     Collator        _____
      Envelope        65                    Face-down       _____
      Large-capacity  ___                   Face-up         _____
      Main            ___                   Large           _____
      Manual          100                   Left            _____
      Middle          ___                   Middle          _____
      Side            ___                   Private         _____
      Top             1                     Right           _____
                                            Side            _____
                                            Top             _____
```

Figure 18. Defining PSF/MVS Printers to the Print Interface (Screen 3 of 3)

2.4.3 Defining General Printers

The Print Interface converts text data into S/390 line data for this type of printer. Therefore, when you define a general printer, it must be able to print S/390 line data.

A general printer definition is for printers not in the other categories, PSF/MVS or IP PrintWay. For general printers, but you can specify the same attributes as

with PSF/MVS and IP PrintWay. The following additional attributes can be specified, as shown in Figure 20 on page 24:

- Forms control buffer (FCB)
- Universal character set (UCS)

Figure 19 and Figure 20 on page 24 show the panel for defining a general printer to the Print Interface.

```
ADD    ---------- OS/390 Print Interface: General Printer Definition ---------
COMMAND ===>                                                 USERID - ROGERS
                                                             DATE   - 98/04/07
  Printer Name  ===> _____________                           TIME   - 18:40
                                                                    More:     +
  Description ===> ____________________________________________________
  ____________________________________________________________________

  Printer Location ===> ___________________________________________
  ____________________________________________________________________

 Work Selection or Routing Selection:
   CLASS     ===> _    DEST         ===> ________   FORMS  ===> ________
   Priority  ===> ___  Process mode ===> ________   Writer ===> ________

   Forms Control Buffer    ===>
   Universal Character Set ===>

   Burster-trimmer-stacker: 3  1. Yes
                               2. No
                               3. Unspecified
 Resources:
   Form Definition  ===> ________
   Page Definition  ===> ________
   Front Overlay    ===> ________
   Back Overlay     ===> ________

 Transmission Attributes:
   Print Queue Name ===> ____________________________________________
   ____________________________________________________________________
```

Figure 19. Defining General Printers to the Print Interface (Screen 1 of 2)

```
   Printer IP Address ===> _____________________________________________
                           _____________________________________________

   Options Entry ===> _________________

   Retry       : Time                  (HHHH:MM:SS)
               : Limit   __________    (0-32767)
   Retain Time : Success __________    (HHHH:MM:SS or FOREVER)
               : Failure __________    (HHHH:MM:SS or FOREVER)

   Destination Node ===> ________

 Code Pages: Document ===> _______________    Printer ===> IBM-1047 ___

 -------------- OS/390 Print Interface: Validation Attributes --------------

   Maximum Job Size:
    1  1. No Limit
       2. Bytes  ===> ____________(1-2147483647)

   Maximum Copies:
    1  1. No Limit
       2. Number ===> _____       (1-255)

   Forms Supported:
    1  1. All Forms
       2. Form ===> ________  ________  ________ ________  ________
 Data Formats Supported and associated Filter Paths:
                                                              ("/" or blank)
   Data Format                       Filter Path
 ============= ============================================================
 / Line-data   _____________________________________________________________
 / MO:DCA-P    _____________________________________________________________
 / PostScript  _____________________________________________________________
 / Text        _____________________________________________________________
 / PCL         _____________________________________________________________
 / Other       _____________________________________________________________
 Duplex Supported:  / Simplex    / Duplex     / Tumble       ("/" or blank)

 Print-error Reporting Supported:  / Character  / Position   ("/" or blank)

 ------ OS/390 Print Interface: Input Tray and Output Bin Attributes ------

 Input Trays:                          Output Bins:

       Name          Number                  Name          Number
       Alternate       1                     Bottom          _____
       Bottom          2                     Collator        _____
       Envelope        65                    Face-down       _____
       Large-capacity  ___                   Face-up         _____
       Main            ___                   Large           _____
       Manual          100                   Left            _____
       Middle          ___                   Middle          _____
       Side            ___                   Private         _____
       Top             1                     Right           _____
                                             Side            _____
                                             Top             _____
```

Figure 20. Defining General Printers to the Print Interface (Screen 2 of 2)

2.4.4 Listing Defined Printers

After the printers are defined or new ones added, Option 2 from the primary panel can be used to display the defined printers.

```
------------------- OS/390 Print Interface: Definition List  Row 1 to 11 of 11
COMMAND ===>                                                  SCROLL ===> CSR
Actions: B-Browse  C-Copy  D-Delete  E-Edit  =-Repeat             Printer
A Printer Name      Type Printer Location  DEST     C FORMS      IP Address
= ================ ==== ================ ======== = ======== ===============
  lpt2              IPPW 2C-16            LPT2     J STD      9.12.14.63
  poke              IPPW 2c16             POKE     J STD      9.12.2.4
  pokej2            IPPW 2C-16            POKEJ2   J STD      9.12.2.4
  pokej3            IPPW 2C-16            POKEJ3   J STD
  pokeps            IPPW 2c16             POKEPS   J STD      9.12.2.4
  prt5              PSF  2c16             POK3130E U STD
  FIIRPS            IPPW IBM 3F1, Helsink FIIRPS   J STD      9.84.244.81
  FIJVBIN           IPPW VAINI IBM 3F1, H FIJVLP   J STD      9.84.252.233
  FIJVLP            IPPW VAINI IBM 3F1, H FIJVLP   J STD      9.84.252.233
  FISLPS            IPPW IBM 3F1, Helsink FISLPS   J STD      9.84.254.207
  I3130P2           IPPW Syslab           I3130P2  J STD      9.12.0.140
****************************** Bottom of data ********************************
```

Figure 21. Print Interface Printers

2.5 Starting the Print Interface

We would recommend that you start the Print Interface by using a started task as shown in 2.5.3, “Running as a Started Task” on page 27 and using the BPXBATCH utility. There are three ways to start the Print Interface:

- From OMVS, issue the AOPSTART command
- Operator issues a start task command, as shown in 2.5.3, “Running as a Started Task” on page 27
- Operator issues a started job command, as shown in 2.5.4, “Running as a Started Job” on page 28

2.5.1 Defining an Environment Variable File

To pass environment variables to BPXBATCH, define a file containing the variable definitions; it can be one of these:

- An HFS file identified with an STDENV statement
- An MVS data set identified with an STDENV statement

The default is /dev/null, which means no variables are defined.

If you define an HFS file:

- It must be a text file defined with read access only.
- Specify one variable per line, in the format VARIABLE=VALUE. Environment variable names must begin in column 1.
- An environment variable file cannot have sequence numbers in it. If you use the ISPF editor to create the file, set the sequence numbers off by typing number off on the command line before you begin typing the data. If

sequence numbers already exist, type UNNUM to remove them and set the number mode off.

If you define an MVS data set:

- It must be a sequential data set, a partitioned data set (PDS) member or a SYSIN data set. Record format should be variable (RECFM=V). Data sets allocated with a fixed format (RECFM=F) tolerate padding with blanks. These blanks are counted when calculating the size of the line and can affect your environment variable settings.
- Specify one environment variable per record, in the format VARIABLE=VALUE. Environment variable names must begin in column 1. Do not use terminating nulls.
- An environment variable file cannot have sequence numbers in it. If you use the ISPF editor to create the file, set the sequence numbers off by typing number off on the command line before you begin typing the data.

Programs that run in the OS/390 C/C++ environment read a set of environment variables. These environment variables can describe things like the command search path, the OS/390 Print Interface Dynamic Link Library (DLL) files, and the current time zone, to name a few. The environment variables that are needed to run the OS/390 Print Interface using BPXBATCH were defined in a file in the HFS. Its contents are shown in Figure 22 on page 27.

AOPCONF Defines the path to the OS/390 Print Interface configuration file.

PATH Defines a set of HFS directories to search when trying to locate an executable.

LIBPATH Defines the absolute or relative path to be searched when loading DLLs for the OS/390 Print Interface. Add /usr/lpp/Printsrv/lib to any existing values.

MANPATH Specifies the path of directories that contain the man pages, which displays help information about a shell command or searches for help files having the specified keywords associated with them. Add /usr/lpp/Printsrv/man/C to any existing values.

NLSPATH Specifies the path of directories that contain message catalog files. Add /usr/lpp/Printsrv/en_US/%N to any existing values.

Note: %N is the catalog file name passed.

Note: The /usr/lpp/Printsrv/man/C directory must occur before /usr/man/%L in the MANPATH environment variable, so that the OS/390 Print Interface version of the lp, lpstat, and cancel man pages is displayed.

2.5.2 Running in Batch Using the BPXBATCH Utility

You may want to execute your OS/390 Print Interface application using JCL in a batch job that executes the BPXBATCH utility. BPXBATCH is an MVS utility that you can use to run shell commands or shell scripts and to run executable files through the MVS batch environment.

We provide here an example of how you can use BPXBATCH to execute your OS/90 Print Interface from JCL, as shown in Figure 22 on page 27. With BPXBATCH, you can allocate the MVS standard files stdin, stdout, and stderr as HFS files. If you do allocate these files, they must be HFS files. You can also allocate MVS data sets or HFS text files containing environment variables

(stdenv). If you do not allocate them, stdin, stdout, stderr, and stdenv default to /dev/null. Allocate the standard files using the data definition PATH keyword options, or standard data definition options for MVS data sets, for stdenv.

```
//PRINTINT JOB ' ','ROGERS',CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
//         REGION=4M,TIME=1440,NOTIFY=&SYSUID
//*****************************************************************
//* RUN THE OS/390 Print Interface FROM BATCH
//*
//*****************************************************************
//STEP1   EXEC PGM=BPXBATCH,
//   PARM='PGM /usr/lpp/Printsrv/bin/aopstart'
//STDOUT   DD PATH='/tmp/Printsrv-stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//STDERR   DD PATH='/tmp/Printsrv-stderr',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//*****************************************************************
//*Define OS/390 Print Interface variables in the HFS
//*****************************************************************
//STDENV   DD *
AOPCONF=/etc/Printsrv/aopd.conf
PATH=/usr/lpp/Printsrv/bin
LIBPATH=/usr/lpp/Printsrv/lib
MANPATH=/usr/lpp/Printsrv/man/C
NLSPATH=/usr/lpp/Printsrv/en_US/%N
/*
//
```

Figure 22. Batch Job JCL to Start the Print Interface

2.5.3 Running as a Started Task

To start the Print Interface from an operator command, you can create a procedure in your PROC library.

We placed the procedure, shown in Figure 23, in member PRINTS in the SYS1.PROCLIB data set.

```
//PRINTS  PROC
//PRINTS  EXEC PGM=BPXBATCH,
//   PARM='PGM /usr/lpp/Printsrv/bin/aopstart'
//STDOUT   DD PATH='/tmp/Printsrv-stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//STDERR   DD PATH='/tmp/Printsrv-stderr',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//STDENV   DD PATH='/etc/Printsrv.envvars',
//          PATHOPTS=ORDONLY
```

Figure 23. Procedure in SYS1.PROCLIB to Start the Print Interface

As you see in the STDENV DD statement and in Figure 24 on page 28, we placed the environment variables in file /etc/Printsrv.envvars.

```
AOPCONF=/etc/Printsrv/aopd.conf
PATH=/usr/lpp/Printsrv/bin
LIBPATH=/usr/lpp/Printsrv/lib
MANPATH=/usr/lpp/Printsrv/man/C
NLSPATH=/usr/lpp/Printsrv/en_US/%N
TZ=EST5EDT
_BPXK_SETIBMOPT_TRANSPORT=TCPIPOE
```

Figure 24. Environment Variables to Be Used with the Print Interface

To start the Print Interface, issue the following operator command:

```
S PRINTS
```

When the task executes, it creates a forked address space as shown in Figure 25.

```
IEE114I 14.21.06 1998.153 ACTIVITY 793
 JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
00010    00023    00006      00028    00099    00006/00030       00023
 LLA      LLA      LLA       NSW  S  JES2     JES2     IEFPROC  NSW  S
 VTAM44   VTAM44   NET       NSW  S  DFSMSHSM HSMSC67  DFSMSHSM NSW  S
 TCPIPOE  TCPIPOE  TCPIP     NSW  SO VLF      VLF      VLF      NSW  S
 RMF      RMF      IEFPROC   NSW  S  APPC     APPC     APPC     NSW  S
 ASCH     ASCH     ASCH      NSW  S  IHV      IHV      PSTEP01  OWT  S
 SDSF     SDSF     SDSF      NSW  S  DFRMM    DFRMM    IEFPROC  NSW  S
 OPTSO    OPTSO    OPTSO     OWT  S  RACF     RACF     RACF     NSW  S
 TSO      TSO      STEP1     OWT  S  INETD1   STEP1    OMVSKERN OWT  AO
 PMAPOE1  STEP1    STC       OWT  AO FTPDOE1  STEP1    FTPDOE   OWT  AO
 MVSNFS5  MVSNFS5  GFSAMAIN  NSW  SO IMWEBRHR IMWEBRHR WEBSRV   IN   SO
 IMWEBKMT IMWEBKMT WEBSRV    IN   SO WEBSRV   STEP1    WEBSRV   OWT  AO
 SUFINIT1 SUFINIT1 *OMVSEX   OWT  SO SUFINIT1 *OMVSEX  STC      IN   AO
 SUFINIT2 SUFINIT2 *OMVSEX   OWT  SO SUFINIT2 *OMVSEX  STC      IN   AO
 IMWEBSUF IMWEBSUF WEBSRV    IN   SO WEBSRV   STEP1    WEBSRV   IN   AO
 RUNCF    RUNCF    IEFPROC   NSW  S  TCPIPMVS TCPIPMVS TCPIP    NSW  SO
 ROGERSG  GO       TSO       OWT  JO FTPDMVS1 STEP1    STC      OWT  AO
 PRINTS1  *OMVSEX  STC       OWT  AO
EMDE    IN   0   KYNEF   OWT      ROGERS  IN       *LOGON* OWT
```

Figure 25. DISPLAY ACTIVITY Command to Show Forked Address Space

2.5.4 Running as a Started Job

To start the Print Interface from an operator command, you may alternatively create a started job JCL in the started job library. This JCL must be placed in a data set referenced in the MSTJCLxx data set in PARMLIB. See 2.5.4.1, “Creating a Started Jobs Data Set” on page 29.

We placed the JCL, shown in Figure 26 on page 29, in member PRINTINT in the started jobs data set SYS1.STCJOBS.

```
//PRINTS   JOB ' ','ROGERS',MSGCLASS=H,MSGLEVEL=(1,1),
//         REGION=0M,TIME=1440
//*****************************************************************
//* RUN AN OE PROGRAM AS A STARTED JOB
//*     S PRINTINT,JOBNAME=PRINTS
//*****************************************************************
//STEP1   EXEC PGM=BPXBATCH,
//   PARM='PGM /usr/lpp/Printsrv/bin/aopstart'
//STDOUT   DD PATH='/tmp/Printsrv-stdout',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//STDERR   DD PATH='/tmp/Printsrv-stderr',
//          PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//          PATHMODE=SIRWXU
//STDENV   DD *
AOPCONF=/etc/aopd.conf
PATH=/usr/lpp/Printsrv/bin
LIBPATH=/usr/lpp/Printsrv/lib
MANPATH=/usr/lpp/Printsrv/man/C
NLSPATH=/usr/lpp/Printsrv/en_US/%N
TZ=EST5EDT
/*
```

Figure 26. JCL from SYS1.STCJOBS to Start the Print Interface

Then the following operator command can be issued:

```
S PRINTINT,JOBNAME=PRINTS
```

When this started job executes, it creates a forked address space as shown in Figure 27.

```
IEE114I 16.15.11 1998.072 ACTIVITY 786
 JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
00004    00017    00003      00028    00093    00003/00030       00010
 LLA      LLA      LLA      NSW  S  JES2     JES2     IEFPROC  NSW  S
 VTAM44   VTAM44   NET      NSW  S  DFSMSHSM HSMSC67  DFSMSHSM NSW  S
 VLF      VLF      VLF      NSW  S  RMF      RMF      IEFPROC  NSW  S
 APPC     APPC     APPC     NSW  S  ASCH     ASCH     ASCH     NSW  S
 SDSF     SDSF     SDSF     NSW  S  DFRMM    DFRMM    IEFPROC  NSW  S
 OPTSO    OPTSO    OPTSO    OWT  S  RUNCF    RUNCF    IEFPROC  NSW  S
 RACF     RACF     RACF     NSW  S  MVSNFS5  MVSNFS5  GFSAMAIN NSW  SO
 PORTMAP5 STEP1    STC      OWT  AO INETD1   STEP1    OMVSKERN OWT  AO
 TCPIPOE  TCPIPOE  TCPIP    NSW  SO FTPD1    STEP1    STC      OWT  AO
 PRINTWA2 PRINTWAY IEFPROC  NSW  SO TSO      TSO      STEP1    OWT  S
 PRINTS1  *OMVSEX  STC      OWT  AO
ROGERS  IN       RALPHR  OWT      ANDREVN OWT
```

Figure 27. DISPLAY Command to Show Forked Address Space

2.5.4.1 Creating a Started Jobs Data Set

To create a job for a started task, you can use a data set that contains only jobs, or you can mix the jobs with procedures. If you want to keep the jobs and procedures separate, you can use the IEFJOBS data set and update your MSTJCLxx to include the IEFJOBS DD as follows:

```
//IEFJOBS  DD DSN=SYS1.STCJOBS,DISP=SHR
```

You can concatenate several data sets to the IEFJOBS ddname, exactly as you can do for the data sets concatenated to the IEFPDSI ddname.

If you do not want to update MSTJCLxx, you can still take advantage of this support by placing the jobs you want to start in one of the IEFPDSI data sets.

However, we recommend that you define the IEFJOBS ddname in MSTJCLxx for several reasons:

- By defining the ddname IEFJOBS, you can keep jobs and procedures separated. This allows jobs and the procedures they invoke to have the same name.
- Existing procedure libraries need not be modified.
- You need not worry about products that ship procedures that are placed in procedure libraries. If you modify a procedure to add the JOB card and a new version of the procedure is received, you do not lose your modifications.

With the introduction of the new IEFJOBS DD, it is important to note the following definitions:

IEFJOBS DD data sets must contain *only* jobs.

IEFPDSI DD data sets can contain both jobs and procedures.

JES JES procedure libraries must contain only jobs.

If the member to be started is found in IEFJOBS and it contains a procedure instead of a job, the START command fails with message IEE404I.

Again, it is advisable to use the IEFJOBS DD in the master JCL to keep those started tasks using jobs as their source separate from those using standard procedures. Procedures should still reside in a library pointed to by the IEFPDSI DD card in the MSTJCLxx member used at IPL time if they are to be started under the control of the master subsystem.

In the example below, we see that the IEFJOBS DD card points to a data set called SYS1.STCJOBS. This name can be any installation-chosen name.

```
//MSTRJCL  JOB MSGLEVEL=(1,1),TIME=1440
//         EXEC PGM=IEEMB860,DPRTY=(15,15)
//STCINRDR DD SYSOUT=(A,INTRDR)
//TSOINRDR DD SYSOUT=(A,INTRDR)
//IEFPDSI  DD DSN=SYS1.PROCLIB,DISP=SHR
//IEFJOBS  DD DSN=SYS1.STCJOBS,DISP=SHR
//SYSUADS  DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC   DD DSN=SYS1.BRODCAST,DISP=SHR
```

2.6 Printing to Print Interface Defined Printers

The following operating systems can be used to send print data sets to the printers defined in Figure 21 on page 25:

- TSO/E on OS/390 systems using the LPR command, see 2.8, “Submitting Print Data Sets From TSO/E” on page 37
- UNIX System Service using the lp command, see 2.7, “Printing from UNIX System Services” on page 34

- OS/2 using the LPR command see 2.9, “Submitting Print Requests from OS/2” on page 38
- VM/CMS using the LPR command
- Windows 95 and Windows NT using OS/390 Print Server clients, see 2.11, “OS/390 Print Server Clients for Windows 95 and Windows NT” on page 39
- AIX 4.1.4 or AIX 4.2.x using the lpr command

2.6.1 Print Interface Processing

Clients can submit print requests from any operating system in the TCP/IP network using the protocol defined in RFC 1179 (see Appendix A, “RFC 1179” on page 187). Clients can also submit print requests from the following operating environments, as shown in Figure 28 on page 33:

- MVS and OS/390 TSO/E
- UNIX System Services (OpenEdition)
- Windows 95 and Windows NT
- OS/2
- AIX
- Win 3.1

Except for Windows 95 and Windows NT, the job submitter can do the following by using the TCP/IP commands as per 2.2.1, “Remote Systems in the TCP/IP LAN Network” on page 11:

- Specify additional job submission parameters
- Hold and release print jobs
- Query the status of print jobs
- Cancel print jobs

2.6.2 Print Data Formats Supported

Print Interface validates the input data format of the user′s data set and then checks if it can perform data transformation. The print administrator can specify the data formats supported by each logical printer in the Validation Attributes, shown in Figure 14 on page 19.

Before performing any type of transformation of data, the Print Interface attempts to determine the data format of the input file. If the attribute document-format is provided by the job, that value is used. Otherwise, the file is sampled and Print Interface attempts to determine the type, as follows:

- If the file is submitted from a non-local client, then Print Interface assumes one of the following data types, depending on the contents of the first characters of the file:
  - ASCII, POSTSCRIPT, PCL, AFPDS, and DBCS-ASCII.
- If the file is submitted from an MVS or UNIX System Services client, Print Interface cannot reliably determine the data type. For instance, distinguishing between ASCII data, mixed-mode line data, and EBCDIC DBCS data is highly error-prone.

If the job submitter does not specify a data format and the Print Interface cannot determine the data format, the data format is set to one of the following defaults:

- If the input file is from a non-MVS client, the default is ASCII.

- If the input file is from an MVS or UNIX System Services client, the default is EBCDIC line-data.

2.6.2.1 Data Formats Printable Without Transformation

Print Interface allows different data formats for different types of printers, with the defaults as follows:

PSF The data formats supported by PSF/MVS are line-data and modca-p.

IP PrintWay All defined formats, such as ASCII, DBCS-ASCII, ISO-6429, line-data, MO:DCA-P, PCL, PostScript, and simple-text.

2.6.3 Print Interface and Print Requests

The Print Interface uses some printer attributes to dynamically allocate data sets to the JES spool. When defining the Print Interface definition table, you must define printer attributes to match the required JES work selection criteria as defined to JES. If class J is a JES work selection specification for a printer or set of printers, then you should specify class J in the Print Interface printer definition. For the dyNAmic allocation request made by the Print Interface, the attributes are obtained from the print entry specified by the submittor. If the print document submittor and the definition table specify the same allocation parameters, the Print Interface uses the document attribute instead of the printer attribute, provided the document attribute is valid for the printer definition.

Beginning with OS/390 Version 2 Release 4, OpenEdition changed from using APPC initiators to WLM controlled initiators. The initiator address space name is BPXAS instead of ASCHINT. BPXAS is a special initiator with special interfaces to WLM and UNIX System Services.

When a print request arrives to the Print Interface from the sources shown in Figure 28 on page 33, Print Interface uses the fork and spawn functions to get an initiator address space. Fork and spawn use a WLM service to get an address space to process the print request. The WLM service checks its queue to see if there is an idle address space from a free pool of BPXAS address spaces that are waiting for work. If there are no idle BPXAS address spaces, WLM determines whether a new one can be created based on system load. If the free pool is empty, an ASCRE is done using the name BPXAS as the procedure to start a new one. When a BPXAS address space is finished, the address space goes back into the free pool. After 30 minutes of being idle, the address space is terminated.

A model SWA is used by the initiator for all requests and is created on the first spawn or fork request. WLM manages the queue of work and invokes a UNIX System Services exit to take care of the attribute propagation from the parent. This is passed back from the exit to allow the SWA to be updated correctly.

Note: For security checking, during fork and spawn processing, the kernel has code that works with the initiator code and security product exits to build the ACEE to match that of the parent.

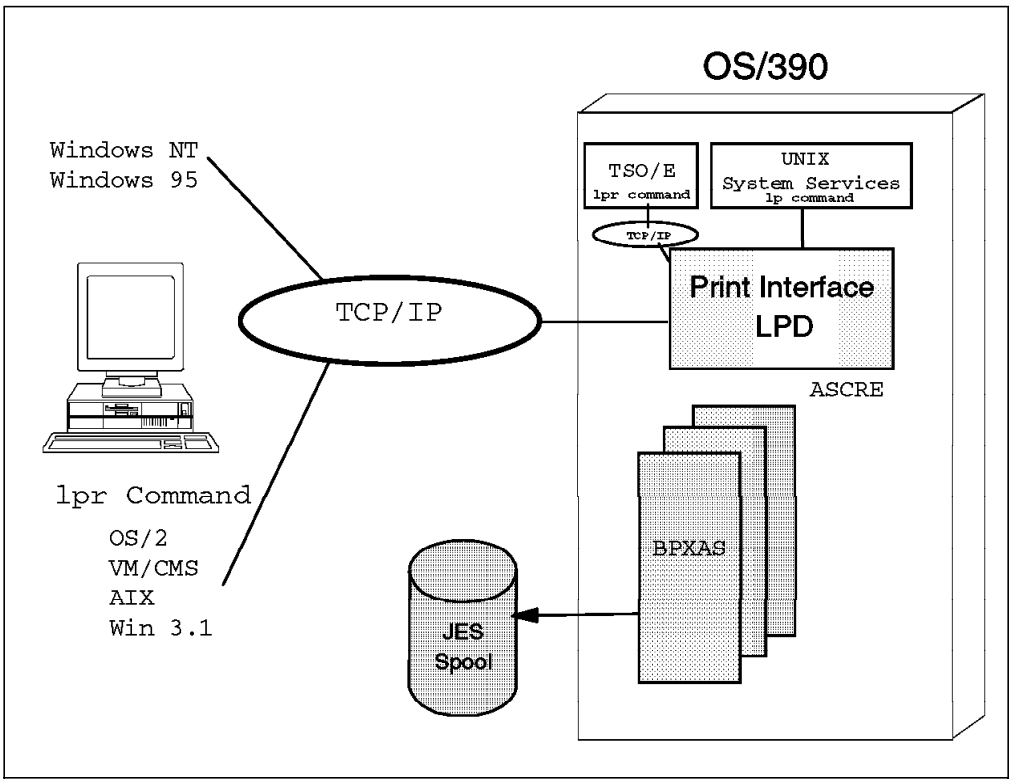


Figure 28. Print Interface Processing Requests

2.6.3.1 Print Interface Requests to JES

The Print Interface maintains information about the status of SYSOUT data sets that it created and uses this to provide status information to Print Server users. To maintain this status information, JES records data provided by the Event Notification Facility (ENF). JES has added a new function code (58) to ENF that enables Print Interface to be informed of changes in the status of SYSOUT data sets that are created. JES issues ENF 58 for the following events:

- Data set purged
- Data set selected
- Data set processed and disposition updated
- Data set no longer selected - disposition not updated
- Data set no longer selected and is held
- Error resulting in a system level hold occurred
- End of data set notification occurred and was successful
- End of data set notification occurred and was unsuccessful
- Job status change occurred (currently only purged)
- Client token has changed

Print Interface also uses Subsystem Interface (SSI) function code 80 (Extended Status) to query the status of SYSOUT data sets that it created.

The status information provided back to Print Server users is the following data set states:

Unknown The data set cannot be located.

Pending The data set has been queued but has not been selected.

Processing The data set has been queued and is selected.

Held The data set has been queued but the job or data set is held.

Retained The data set has been printed and is retained because the data set's retention period has not yet been reached. This state applies only to SYSOUT data sets printed by IP PrintWay.

2.7 Printing from UNIX System Services

The `lp`, `lpstat`, and `cancel` commands, shown in Figure 29, use TCP/IP protocol to send print requests to the OS/390 Print Interface. They send commands to the port number specified in the OS/390 Print Interface configuration file, shown in Figure 4 on page 13.

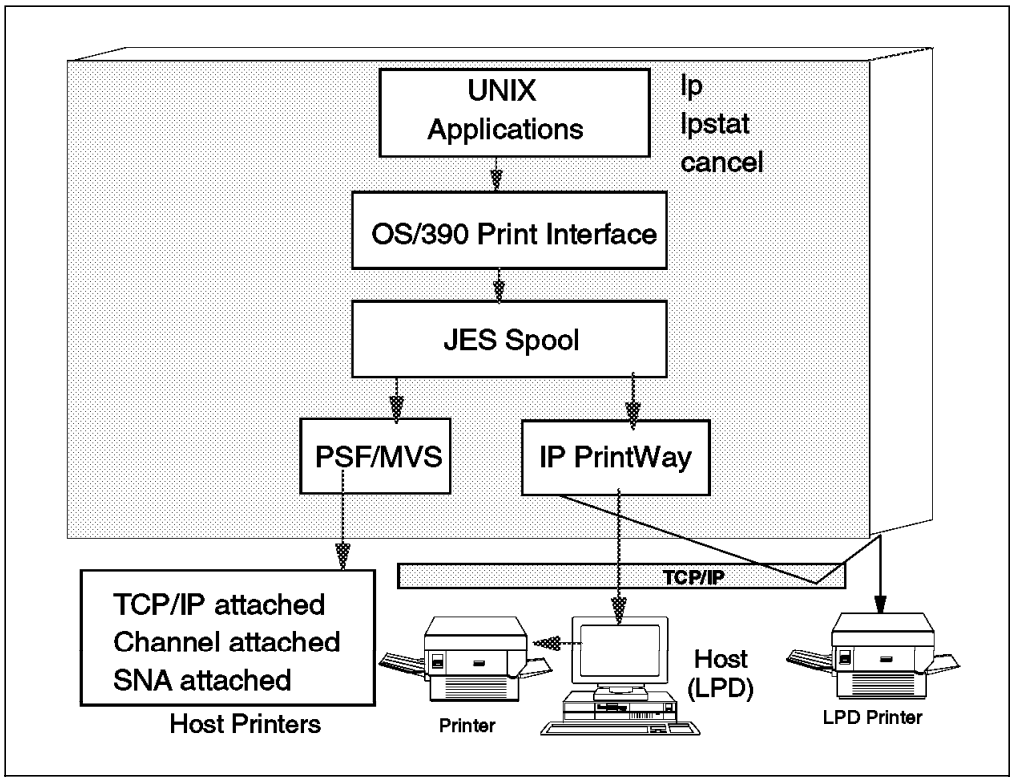


Figure 29. UNIX System Services and OS/390 Print Interface

The `lp`, `lpstat`, and `cancel` commands are modified to be used with the Print Interface and are placed in the HFS as follows:

```
/usr/lpp/Printsrv/bin
```

Make sure that the correct path to the commands is in /etc/profile as follows:

```
PATH=/usr/lpp/Printsrv/bin:/bin:
```

2.7.1 Printers Defined to Print Interface

When the UNIX user needs to know which printers are defined for use, the `lpstat` command can be used as follows:

```
lpstat -a          -   Query names and locations of all printers
lpstat -d          -   Query default printer
lpstat -p poke     -   Query specified printer
lpstat -t          -   Query all printers and jobs
lpstat -u rogers   -   Query all printers and jobs by user ID
lpstat -o poke     -   Query specified printer and jobs
```

and as shown in Figure 30.

```
ROGERS @ SC67:/>lpstat -a
     Printer       Jobs      Location                 Description
---------------- -----  ---------------- ------------------------------
lpt2                 0  2C-16            4029 in 2C-16
poke                 0  2c16             3130 in 2c16
pokeps               0  2c16             3130 in 2c16
prt5                 0  2c16             AFP Printer 5 located in 2C16
FIIRPS               0  IBM 3F1, Helsink IP PrintWay
FIJVBIN              0  VAINI IBM 3F1, H IP PrintWay
FIJVLP               0  VAINI IBM 3F1, H IP PrintWay
FISLPS               0  IBM 3F1, Helsink IP PrintWay
I3130P2              0  Syslab           Syslab 3130
```

Figure 30. Example of the lpstat Command from UNIX System Services

2.7.2 Printing to Print Interface Printers

UNIX System Services users can use the lp command to print data sets to printers defined to the Print Interface. The UNIX user issues the lp command, as shown in Figure 31, specifying *poke* as the Print Interface defined printer shown in Figure 30. The data set to be printed is the MVS data set TEST.JCL.

```
ROGERS @ SC67:/>lp -d poke //test.jcl
AOP007I Job 284 successfully spooled to poke.
ROGERS @ SC67:/>


 ===>
```

Figure 31. UNIX User Submitting a Print Request

Figure 32 on page 36 shows two data sets passed to the Print Interface using the lp command. The JES2 SDSF display shows the following for the first data set displayed:

ROGERS ROGERS is a user ID who has a UID of 0 in the RACF database.

Note: With UNIX System Services, this field is unpredictable.

PS000284 This is the jobid assigned by JES when the BPXAS address space allocates the data set and JES places the data set on the JES spool.

Note: PS is the default job prefix specified in the configuration file as shown in Figure 4 on page 13.

ROGERS ROGERS is the user ID of the submitter.

144 This is the default priority for the data set. If the Print Interface table for the designated printer had a priority in the entry, that priority would have been used.

J This is the SYSOUT class assigned to the data set, taken from the Print Interface table definition for the printer poke.

STD This is the forms which is taken from the Print Interface definition for the printer poke.

POKE The destination of the printer specified by the user in the `lp` command.

```
  Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------
SDSF OUTPUT ALL CLASSES ALL FORMS     LINES 37         LINE 1-1 (1)
COMMAND INPUT ===>                                      SCROLL ===> HALF
PREFIX=*  DEST=POKE  OWNER=*
NP   JOBNAME  JOBID    OWNER    PRTY C FORMS    DEST                TOT-REC
     ROGERS   PSO00284 ROGERS    144 J STD      POKE                     37
```

Figure 32. JES2 SDSF Display of Print Requests From a UNIX User

If the user specifies a printer that is not defined to the Print Interface, the following message, *AOP001E*, is issued:

```
ROGERS @ SC67:/>lp -d pokd //test.jcl
lp: AOP001E Printer pokd is not defined.
ROGERS @ SC67:/>
 ===>
```

Figure 33. Printing to an Undefined Printer Message

2.7.2.1 Query Status of Submitted Jobs

To determine which jobs have been submitted to each printer, specify:

```
lpstat -t
```

as shown in Figure 34 on page 37. This command shows all the printers defined to the Print Interface and all the jobs queued to the Print Interface printers. The Status is explained in 2.6.3.1, “Print Interface Requests to JES” on page 33 and the Format is explained in 2.6.2.1, “Data Formats Printable Without Transformation” on page 32.

```
     Printer       Jobs      Location                  Description
---------------- -----  ---------------- ----------------------------
lpt2                  0  2C-16            4029 in 2C-16
poke                  4  2c16             3130 in 2c16
pokeps                0  2c16             3130 in 2c16
prt5                  0  2c16             AFP Printer 5 located in 2C16
FIIRPS                0  IBM 3F1, Helsink IP PrintWay
FIJVBIN               0  VAINI IBM 3F1, H IP PrintWay
FIJVLP                0  VAINI IBM 3F1, H IP PrintWay
FIJVNBIN              0  VAINI IBM 3F1, H IP PrintWay
FISLCC                0
FISLPS                0  IBM 3F1, Helsink IP PrintWay
FISLPSNB              0
ITSO3130              0
I3130P2               0  Syslab           Syslab 3130

Printer: poke
 Job   Owner    Status   Format   Size    File
----- -------- --------- ------ --------  ------------------------------
  285 ROGERS   pending   text       3149  ROGERS.TEST.JCL
  286 ROGERS   pending   text       3109  ROGERS.TEST.JCL
  287 TCPIPOE  pending   text       2960  //test.jcl
  288 pc-user  pending   text       2997  test.jcl
  289 ROGERS   pending   text       3108  TEST.JCL
  290 ROGERS2  pending   pcl       20902  ... About the IBM AFP Printer"
  291 ROGERS2  pending   pcl       19019  Printing "Options Dialog"
  296 ROGERS   pending   text       3109  ROGERS.TEST.JCL
  297 ROGERS2  pending   pcl       41436  readme95 - Notepad
  298 ROGERS2  pending   pcl       41436  readme95 - Notepad
  311 ALCIDES  pending   text       2960  //test.jcl

Printer: pokeps
 Job   Owner    Status   Format   Size    File
----- -------- --------- ------ --------  ------------------------------
  292 ROGERS2  pending   modca     19422  scop.AOI
  293 ROGERS2  pending   pcl       41436  readme95 - Notepad
  294 ROGERS2  pending   pcl       41436  readme95 - Notepad
  295 ROGERS2  pending   pcl       41436  readme95 - Notepad
  299 ROGERS2  pending   modca      3650  word.AOI
  301 ROGERS2  pending   modca     19506  word.AOI
  302 ROGERS2  pending   modca     19506  word.AOI
  304 ROGERS2  pending   modca     19422  word.AOI
ROGERS @ SC67:/> lpstat -t
 ===>
```

Figure 34. UNIX User Issues lpstat -t Command

2.8 Submitting Print Data Sets From TSO/E

To submit print data sets to the Print Server from a TSO/E session, the following LPR command sends the data set TEST.JCL to printer *poke*, which is defined in the Printer Interface definition table shown in Figure 21 on page 25.

```
lpr test.jcl (HOST 9.12.2.28 PRINTER poke
```

In a sysplex, the Print Interface may be running on only one of the systems. The TSO/E user may be logged on to any of the systems in the sysplex and submit print data sets to the Print Interface.

2.8.1 JES2 SDSF for Print Data Sets From TSO/E

Figure 35 shows the data set sent to the Print Interface for printing by an IP PrintWay printer.

```
  Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF OUTPUT ALL CLASSES ALL FORMS     LINES 37         LINE 1-1 (1)
COMMAND INPUT ===>                                      SCROLL ===> HALF
PREFIX=*  DEST=POKE  OWNER=*
NP   JOBNAME  JOBID    OWNER    PRTY C FORMS    DEST                TOT-REC
     ROGERS   PSO00285 STC       144 J STD      POKE                      2
```

Figure 35. JES2 SDSF Display of Data Sets from TSO/E LPR Command

2.8.2 JES3 (E)JES for Print Data Sets From TSO/E

```
  Navigate  View  Options  Filters  Help
------------------------------------------------------------------------------
WRITER   321S  1J  0T  4 Records (0 Sched)                        Row 1 of 2
Command ===>                                                   Scroll ===> PAGE
Cmd Jobname  Job-ID   JP Status   Process  Comp C Pri #OSE Dest     Records
--- -------- -------- -- -------- -------- ---- - --- ---- -------- ---------->
    BPXAS    STC15635 15 X-SC43                J  15    1 POKE              2
                                               J  15    1 POKEJ2            2
****************************** Bottom of Data ************************
```

Figure 36. JES3 (E)JES Display of Data Sets from TSO/E LPR Command

2.9 Submitting Print Requests from OS/2

From an OS/2 workstation, a user can use the LPR command to send print requests to the Print Interface, as shown in Figure 37. This command sends an AFP document to a PSF/MVS defined Print Interface printer, as shown in 2.4.2, “Defining PSF/MVS Printers” on page 20.

```
[D:\NOTES\DATA]lpr -p prt5 -s 9.12.2.28 sg242070.afp
```

Figure 37. Command to Print an AFP Document from OS/2 Workstation

Printing from an OS/2 workstation requires TCP/IP Version 3. The JES2 SDSF display of the data set on the spool is shown in Figure 38 on page 39.

```
  Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF OUTPUT ALL CLASSES ALL FORMS     LINES 37         LINE 1-1 (1)
COMMAND INPUT ===>                                          SCROLL ===> HALF
PREFIX=*  DEST=POKE  OWNER=*
NP   JOBNAME  JOBID    OWNER    PRTY C FORMS    DEST                TOT-REC
     PC#USER  PS000288 STC       144 J STD      POKE                      2
```

Figure 38. JES2 SDSF Display of Data Sets from TSO/E lpr Command

2.10 OS/390 Print Server Printing from VM

To print from a VM system to the Print Interface, use the LPR command, as shown in Figure 39. This command shown prints data set (test jcl a) to printer poke at host 9.12.2.28, where the Print Interface is active.

```
    lpr test jcl a (p poke at 9.12.2.28
```

Figure 39. Command to Print from a VM/CMS Session

```
  Display  Filter  View  Print  Options  Help
------------------------------------------------------------------------------
SDSF OUTPUT ALL CLASSES ALL FORMS     LINES 37         LINE 1-1 (1)
COMMAND INPUT ===>                                          SCROLL ===> HALF
PREFIX=*  DEST=POKE  OWNER=*
NP   JOBNAME  JOBID    OWNER    PRTY C FORMS    DEST                TOT-REC
     ROGERS   PS000289 STC       144 J STD      POKE                      2
```

Figure 40. JES2 SDSF Display of Data Sets from VM/CMS LPR Command

2.11 OS/390 Print Server Clients for Windows 95 and Windows NT

The OS/390 Print Server provides three client applications that run on Windows 95 and Windows NT. These applications provide the ability to print on OS/390 system. The three Clients, shown in Figure 41 on page 40, are:

- OS/390 Print Server Port Monitor

  See 2.12, “OS/390 Print Server Port Monitor” on page 40

- IBM AFP Printer Driver

  See 2.13, “IBM AFP Printer Driver” on page 49

- IBM AFP Web Browser Plug-in Viewer

  See 2.14, “IBM AFP Plug-in Viewer” on page 58

The OS/390 Print Server Clients for Windows 95 and Windows NT can be downloaded from the OS/390 host. They are located and stored on OS/390 as self-extracting files in directory:

```
/usr/lpp/Printsrv/win/en_US/
```

Alternatively, you can download the latest software from the web using the following site:

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

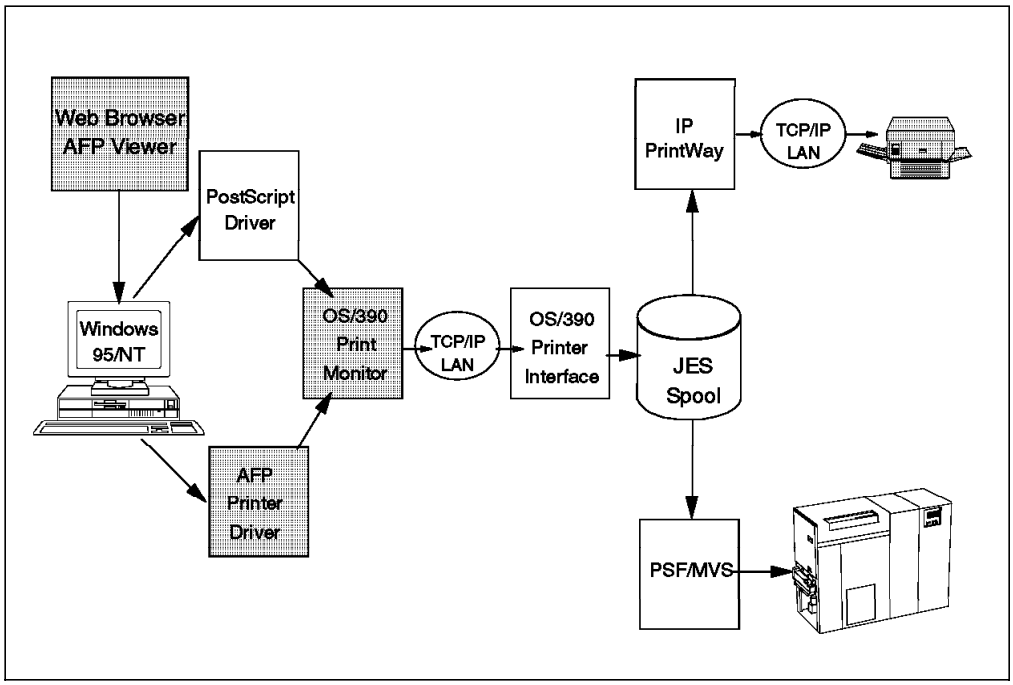


Figure 41. The AFP Printer Driver and OS/390 Print Monitor Overview

2.11.1 Requirements for Using the IBM-Supplied Clients

The IBM-supplied AFP clients require Windows 95 or Windows NT (Version 3.51 or later).

The OS/390 Print Server Port Monitor requires that Microsoft TCP/IP protocol be configured and operational.

The IBM AFP Plug-In Viewer requires Netscape Navigator (Version 3.01 or later) or Microsoft Internet Explorer (Version 3.01, Level 4.70.1215 or later).

2.12 OS/390 Print Server Port Monitor

The Print Server Port Monitor provides the ability to print from any Windows application to a printer that has been defined to the OS/390 Print Interface. The printer can be directly attached to the OS/390 system or can be a printer that is attached to a TCP/IP Local Area Network (LAN).

2.12.1 Printing from Windows 95 and Windows NT

Users can submit jobs for printing using the standard methods of print submission available to Windows applications. A Windows client passes a job with its document attributes to the OS/390 Print Interface. Jobname and owner information are assigned by the Print Interface before the job is routed to the JES spool and then to the subsequent printer either via IP Printway or PSF/MVS, as shown in Figure 41 on page 40. The jobname is the same as the user's LAN ID.

Note: The OS/390 Print Interface does not return error messages or other job process notifications to these clients. In addition, Windows clients cannot query the status of print requests or cancel a print request.

2.12.2 OS/390 Printer Port Monitor Installation

To install the OS/390 Print Server Port Monitor, do the following:

1. Download the file AOPWIN.EXE into a temporary directory.
2. Make a directory that can contain the extracted files:

   ```
   md c:\afpwin
   ```

3. Run the AOPWIN program to extract the files into the directory:

   ```
   c:\temp\aopwin c:\afpwin
   ```

4. Run the setup program to install the port monitor:

   ```
   c:\afpwin\setup
   ```

   The setup program prompts the user for the destination directory during the installation of the product. Use the same directory:

   ```
   c:\afpwin
   ```

5. Restart the Windows system.

2.12.3 Defining Printers for Windows Users

Before selecting a host-defined printer, the following information is required:

- The host name or IP address where the OS/390 Print Server is running.
- The port number of the OS/390 Print Interface deamon, which has a default of 515. This number is the one specified in the configuration file shown in Figure 4 on page 13.
- The name of the printer to use. The system administrator has defined this printer to the OS/390 Print Interface, as shown in Figure 21 on page 25.
- The name of the printer driver to associate with the printer.

Use the standard Windows procedure to add a printer. The example shown is for a Windows 95 system and you should follow these special steps:

1. From the Settings, select **Printers** and then select **Add printer** and the Add Printer Wizard appears as shown in Figure 42 on page 42.

Figure 42. Windows 95 Add Printer Wizard-First Screen

2. Begin installing by selecting **Next** and Figure 43 is displayed.

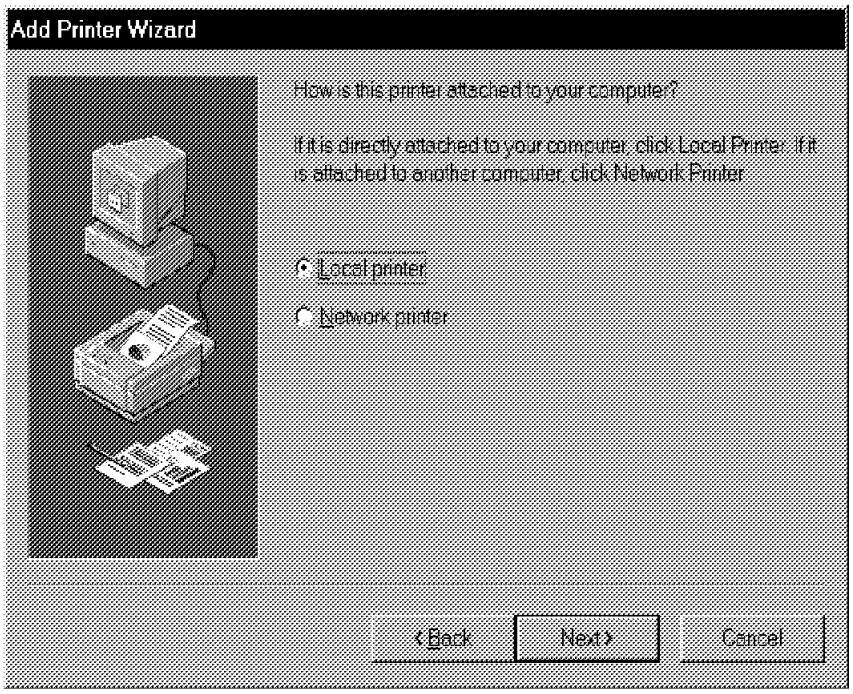


Figure 43. Windows 95 Add Printer Wizard-Second Screen

A Windows administrator can define OS/390 printers as shared printers in the Windows network by selecting Network printer in Figure 43. This allows other Windows users to access the OS/390 Port Monitor, bypassing the need

to have it installed on all the systems in the network. These users can follow standard Windows procedures to add a network printer.

3. To define the printer to the local workstation, select Local printer as shown and the next window contains printers by manufacturers and printer types as shown in Figure 44.

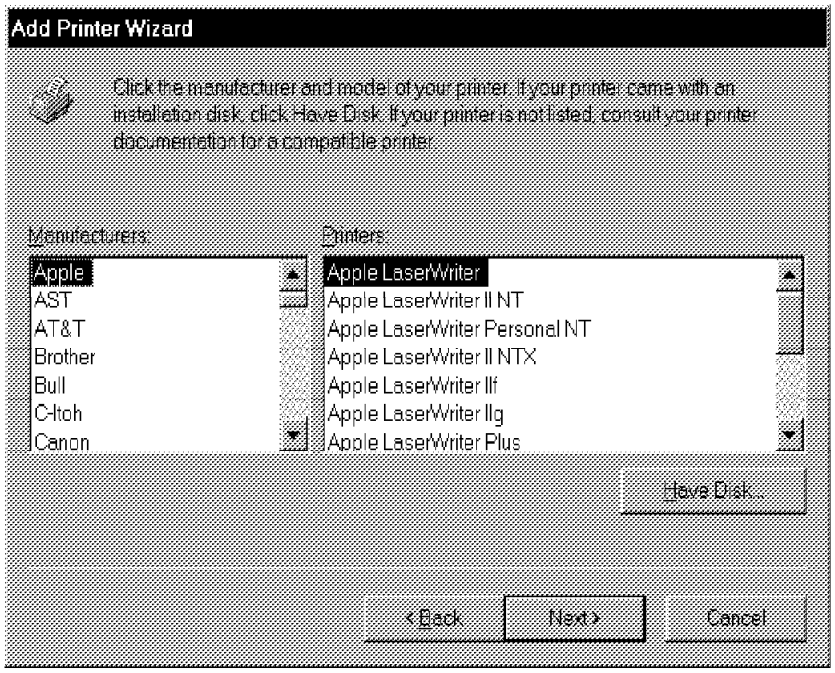


Figure 44. Windows 95 Add Printer Wizard, Select Printer Type

4. Select **Have Disk** which then displays Figure 45 where you can select a printer driver to be loaded.

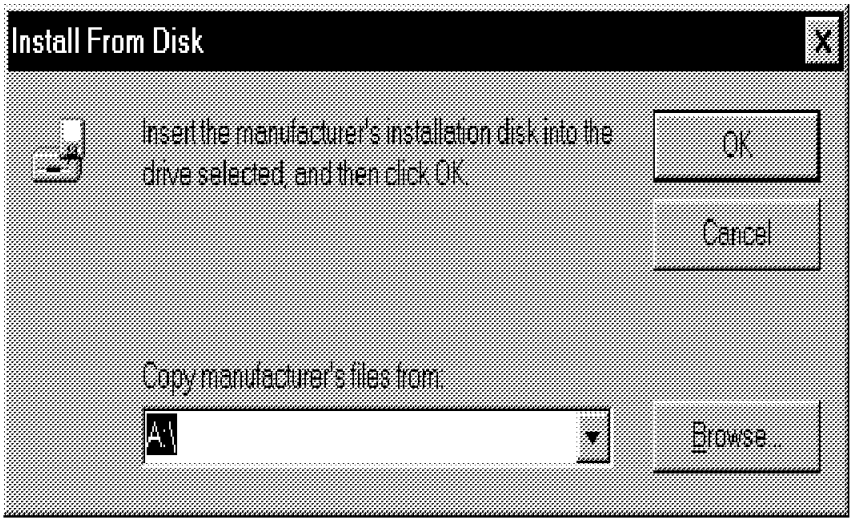


Figure 45. Windows 95 Add Printer Wizard, Load Printer Driver

5. You have a choice to load the printer driver by using a diskette or CD-ROM or any location where the driver exists and then click **OK**.

We selected a diskette on the A:\ drive and it displayed the IBM 3130 print drivers as shown in Figure 46 on page 44.

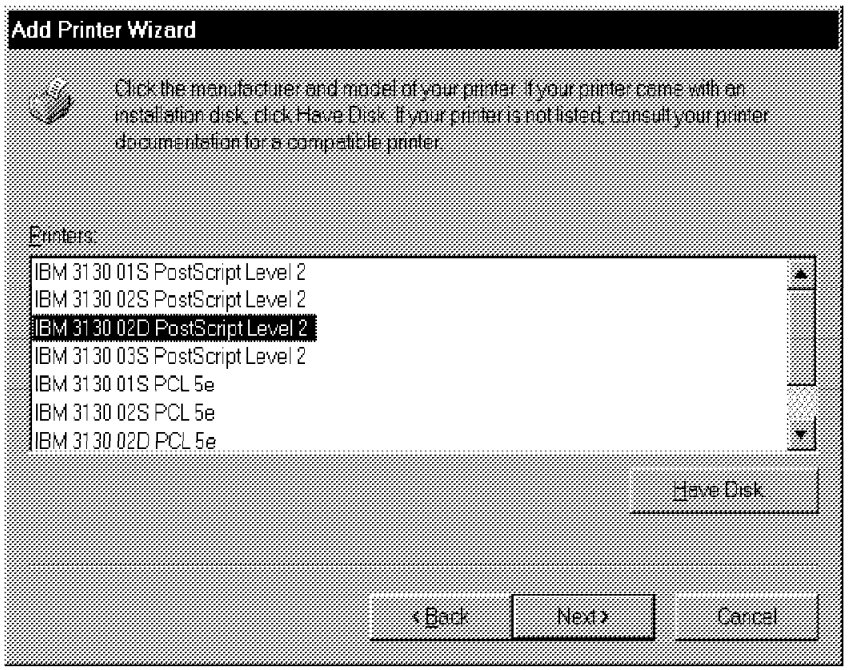


Figure 46. Windows 95 Add Printer Wizard, Select Printer from List

6. Select a printer driver that matches the actual printer you are printing to, as shown in Figure 46, then click **Next** and Figure 47 appears.

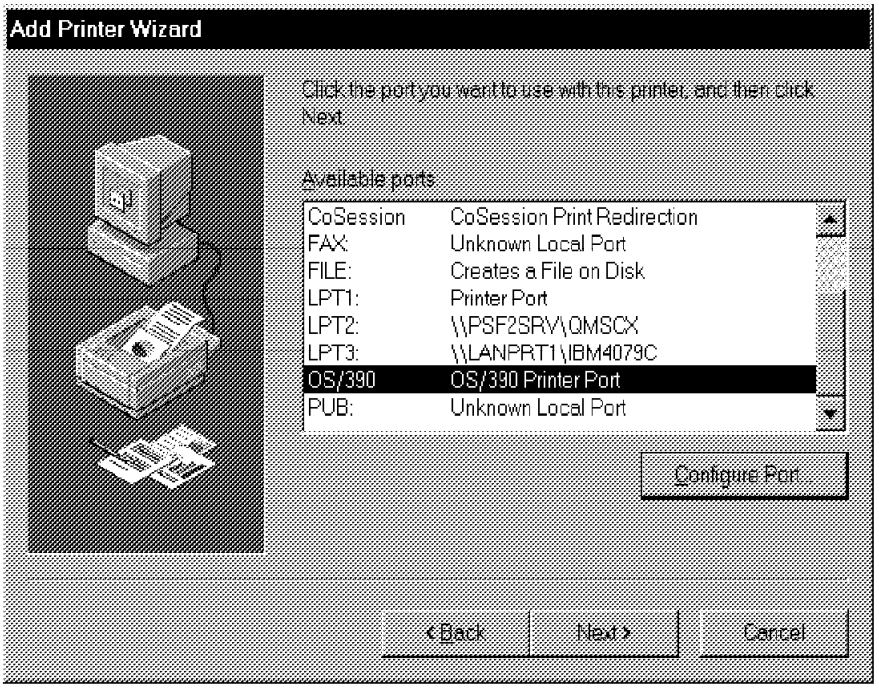


Figure 47. Windows 95 Add Printer Wizard, Select Printer Port

7. Now you should select the **OS/390 Printer Port** and then select **Configure Port** which then gives you Figure 48 on page 45.

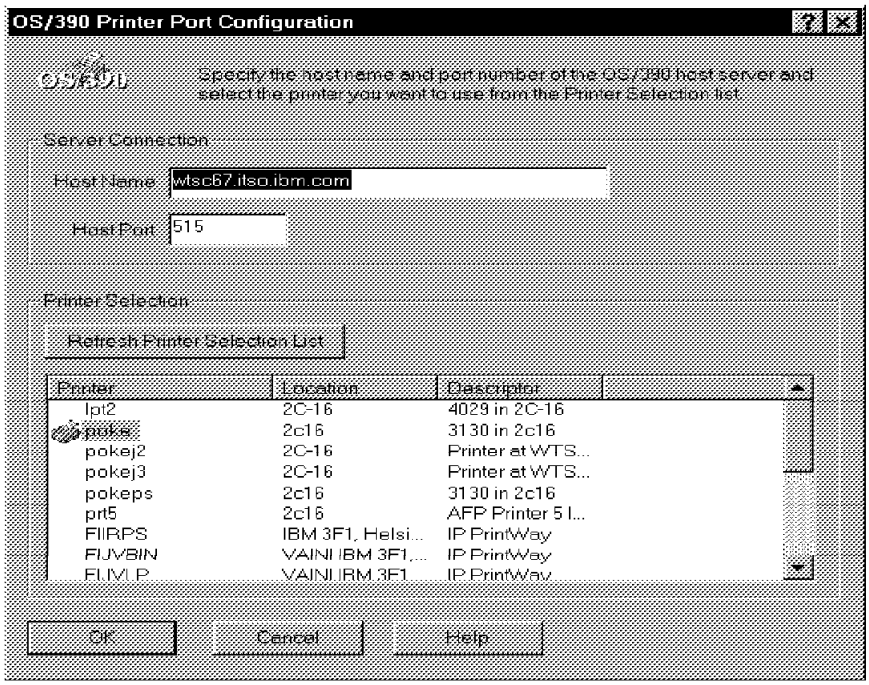


Figure 48. Windows 95 Add Printer Wizard, Select Host Printer

8. Specify a host name, as shown in Figure 48, or an IP address of the host and the Host Port number. Then click **Refresh Printer Selection List** and the Print Interface printer table is passed from the host to the window. Then select the printer from the list you are adding, poke shown in Figure 48. Then click **OK** and Figure 49 on page 46 appears.

Figure 49. Windows 95 Add Printer Wizard, Type Name of Printer

9. Overtype the name of the printer just selected in Figure 48 on page 45 and select **Next** and the last window appears as shown i Figure 50.

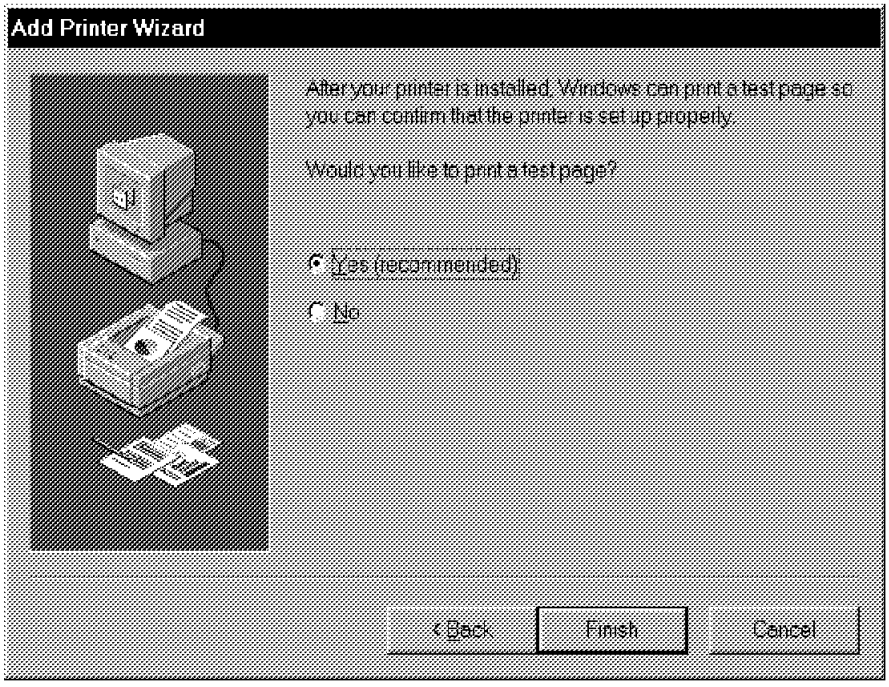


Figure 50. Windows 95 Add Printer Wizard, Select Test Page and Finish

10. Print a test page if desired and select **Finish**. At this point the printer driver is loaded and the printer icon is added to the Add Wizard box. You are now ready to print to the OS/390 printer from an application.

2.12.3.1 Add Wizard With Windows NT

At step 6 shown previously, a different window is shown with Windows NT. When the Add Printer Wizard Window appears, the user can either add or configure a printer port. Figure 51 shows an example of this window where two printers have already been defined. The user can either add an additional printer or elect to configure an existing printer definition.

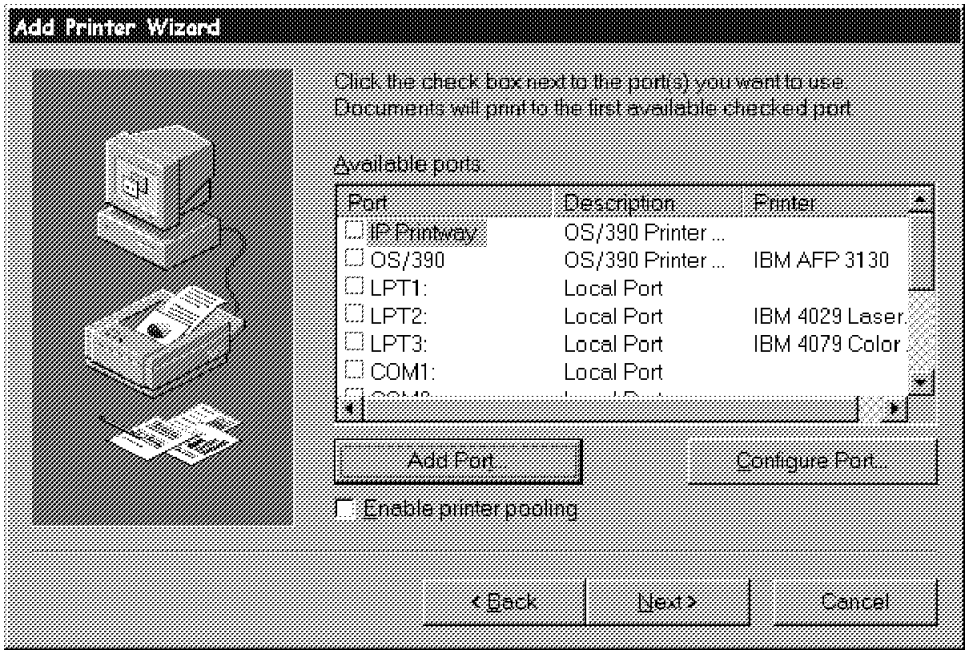


Figure 51. Add Printer Wizard Window for Windows NT for OS/390 Printers

When selecting **Add Port**, the window in Figure 52 is displayed, which lists the available printer types. Select **OS/390 Printer Port**.

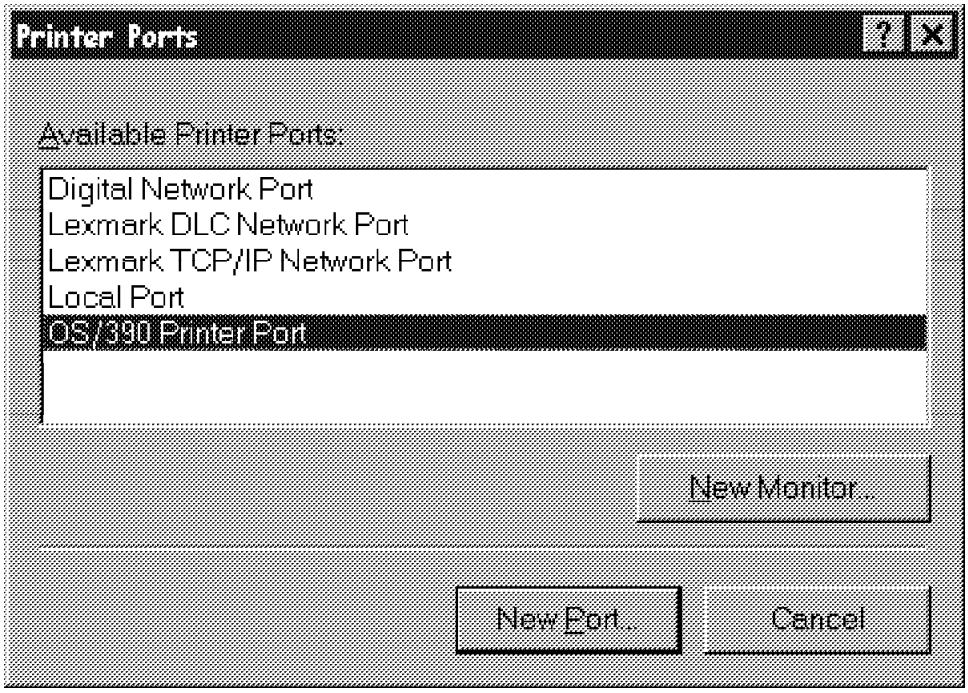


Figure 52. The Available Printer Ports Window for OS/390 Printers

After the port has been added to the available ports list, the printer can be configured. Selecting Configure Port displays the OS/390 Printer Port Configuration window. Use this window to set up the connection between your workstation and the OS/390 host server and to select an OS/390 printer. Host Name Specifies the host name or IP address of the OS/390 host server. The format of the IP address is xxx.xxx.xxx.xxx. Host Port specifies the port number of the Print Server daemon that is running on the OS/390 host server. The default port number is 515. Printer Selection specifies the name, location, and description of each printer that is attached to the OS/390 host server.

Do the following to connect to the OS/390 host server and select the OS/390 printer:

1. Type the host name or IP address in the Host Name entry field.
2. Make sure that the port number specified in the Port entry field is 515 (this is the default).

   Note: Type the port number in the Port entry field if the default port is not appropriate.
3. Click **Refresh Printer Selections** to get a new list of printers.
4. Select the printer you want to use.
5. Click **OK** to return to the Add Printer Wizard window.

   To select a second printer on the OS/390 host system, use the Add Printer wizard to add another OS/390 port and configure it.

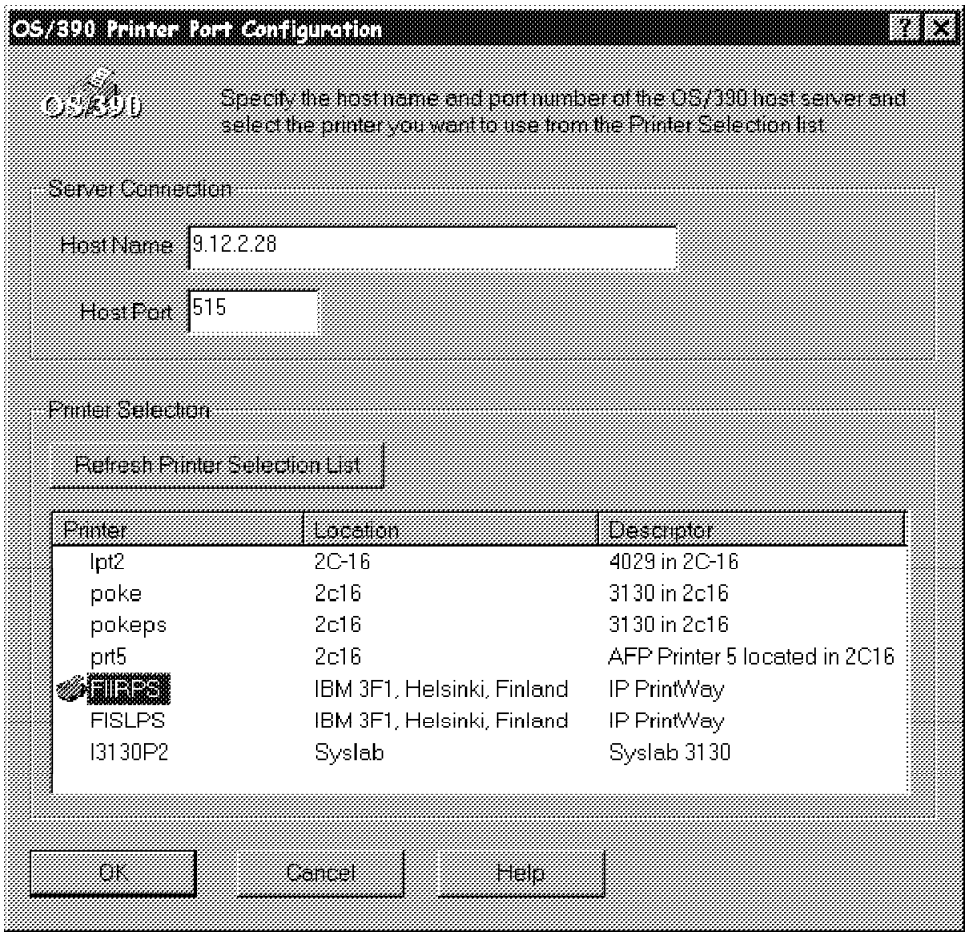


Figure 53. The OS/390 Printer Port Configuration Window

2.13 IBM AFP Printer Driver

The IBM AFP Printer Driver allows users to print from Windows applications to AFP printers defined to JES. Using the IBM AFP printer driver for Windows NT and Windows 95 you can:

- Create AFP documents, overlays, and page segments from your Windows applications.
- Substitute any AFP character set for any of the fonts you use in Windows, so that your text prints as text rather than image.
- Use gray scales color images.
- Use custom paper sizes.
- Use color text (blue, red, pink, green, cyan, yellow, brown).
- Clip page segments and overlays to the offset and size (or printable area) given on the Clip Limit dialog window.

2.13.1 Installing the IBM AFP Printer Driver

To install the OS/390 AFP Pinter Driver, do the following:

1. Download the file AFPDRV95.EXE for Windows 95 or AFPDRVNT.EXE for Windows NT into a temporary directory. The location of these files is specified in 2.12.1, “Printing from Windows 95 and Windows NT” on page 41.
2. Make a directory that can contain the extracted files:

   ```
   md c:\afpdrvr:
   ```

3. Run the AFPDRV95 or AFPDRVNT program to extract the files into the directory:

   ```
   c:\temp\afpfrv95 c:\afpdrvr
   or
   c:\temp\afpfrvnt c:\afpdrvr
   ```

2.13.2 Defining Printers for AFP Printing

The following printers should be defined for AFP printing from a Windows 95 or Windows NT workstation:

1. To be able to create documents in AFP format, you need to define an AFP printer that uses the AFP Printer Driver you have downloaded.
2. You need to define as many printers for the workstation as you need to match the AFP printers available on the host.

2.13.2.1 Define a Printer to Create AFP Output

For option 1, you need to define a printer that prints to a file. To do this, do the following steps with Windows 95:

1. Open the Printers folder and select the **Add Printer** icon. The Add Printer Wizard appears, as shown in Figure 42 on page 42.
2. Select **Next**. Two radio buttons (Local printer and Network printer) are displayed, as shown in Figure 43 on page 42.
3. Select the Local printer radio button, and then select **Next** and Figure 54 on page 50 appears.

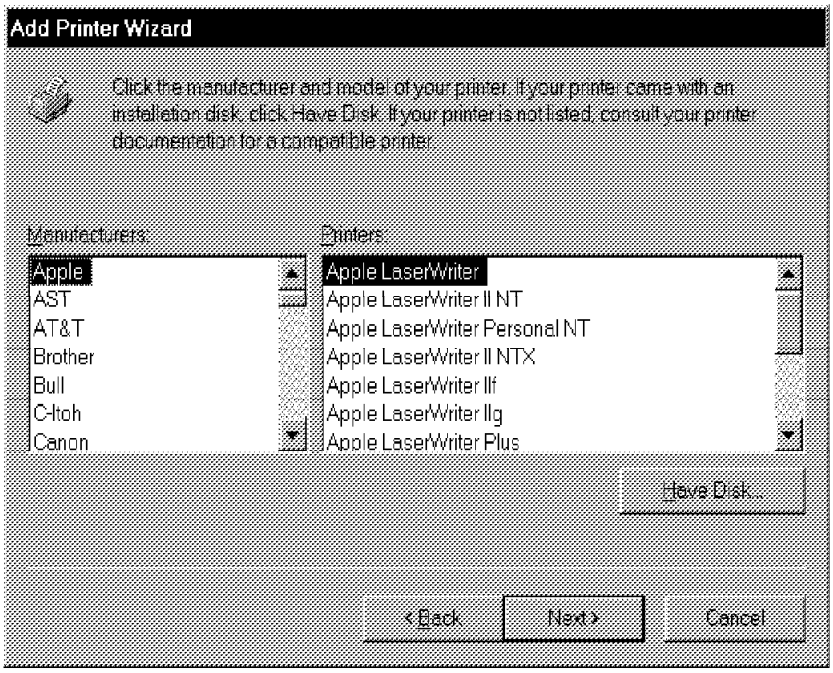


Figure 54. Windows 95 Add Printer Wizard, Select Printer Manufacturer

4. Then you really only need to click **Have Disk** and Figure 55 appears. Selecting a manufacturer does not make any difference.

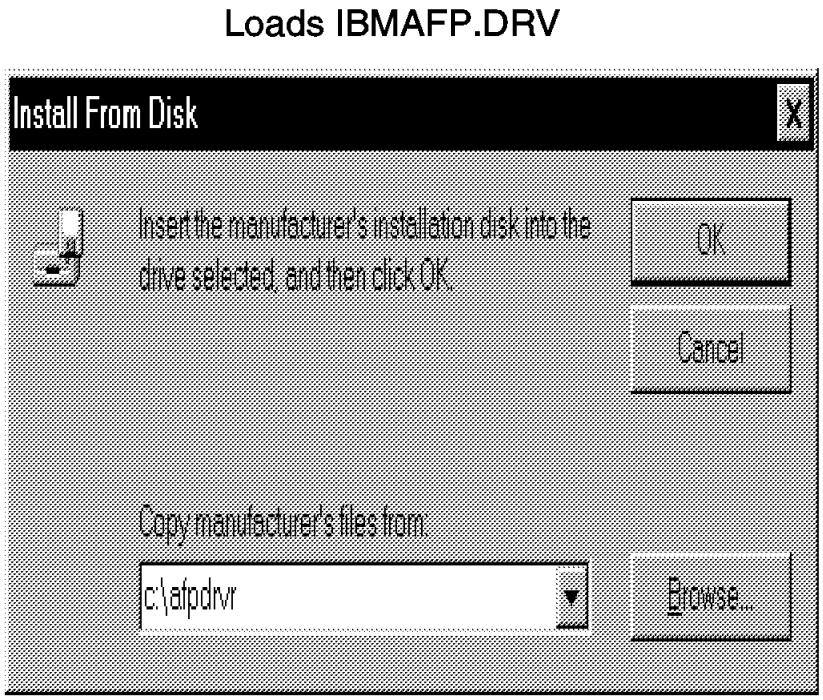


Figure 55. Windows 95 Add Printer Wizard, Select Printer Driver Location

5. You want to install the driver that has been downloaded to your workstation in directory afpdrvr, as now shown in Figure 55. When you Click OK, Figure 56 on page 51 appears.

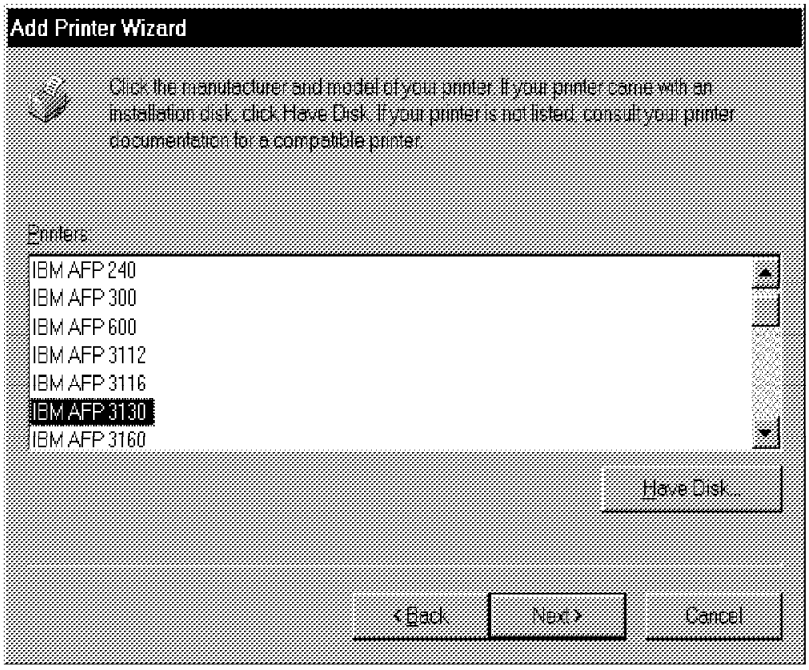


Figure 56. Windows 95 Add Printer Wizard, Select AFP Printer Type

6. Select the printer driver you want to use from the list that is displayed in Figure 56, then select **NEXT**. Figure 57 appears for selection of a printer port.

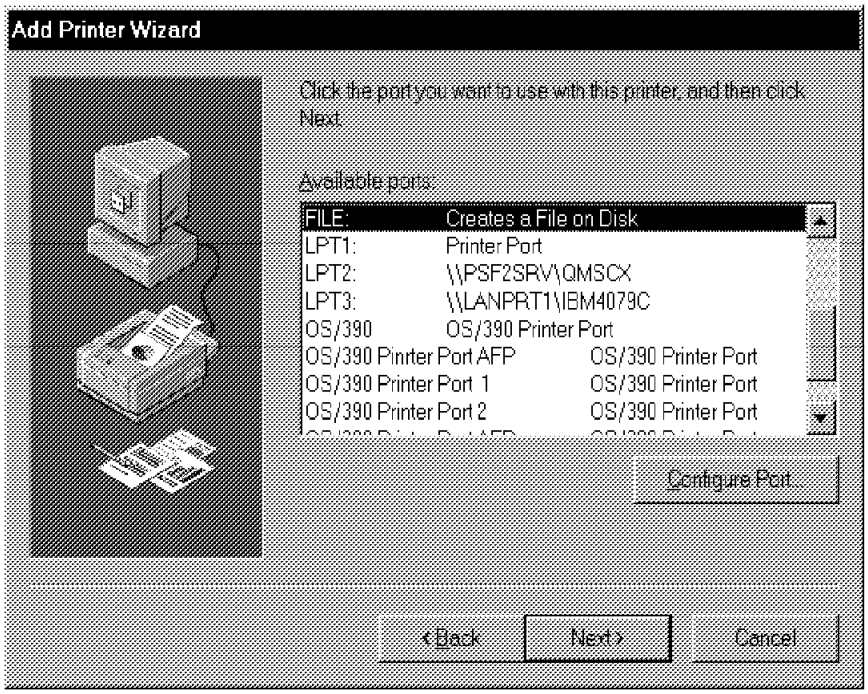


Figure 57. Windows 95 Add Printer Wizard, Select Printer Port

At this point in the Add Wizard dialog, every step is the same for the two options discussed in 2.13.2, “Defining Printers for AFP Printing” on page 49. At this point select:

a. For Option 1, to save the output from the AFP Printer Driver in a file for uploading to a host system or for viewing with the IBM AFP Viewer, select **File**.

b. For Option 2, to print to a PSF/MVS-controlled printer on an OS/390 system, select an OS/390 Printer Port, and then select **Configure Port** to connect to the OS/390 system and specify the name of the OS/390 printer. See 2.13.2.2, “Define a Printer to Print AFP Output on Host” on page 53 for the continuation of this option.

7. Using Option 1 to Select File, then click **Next** and Figure 58 appears.

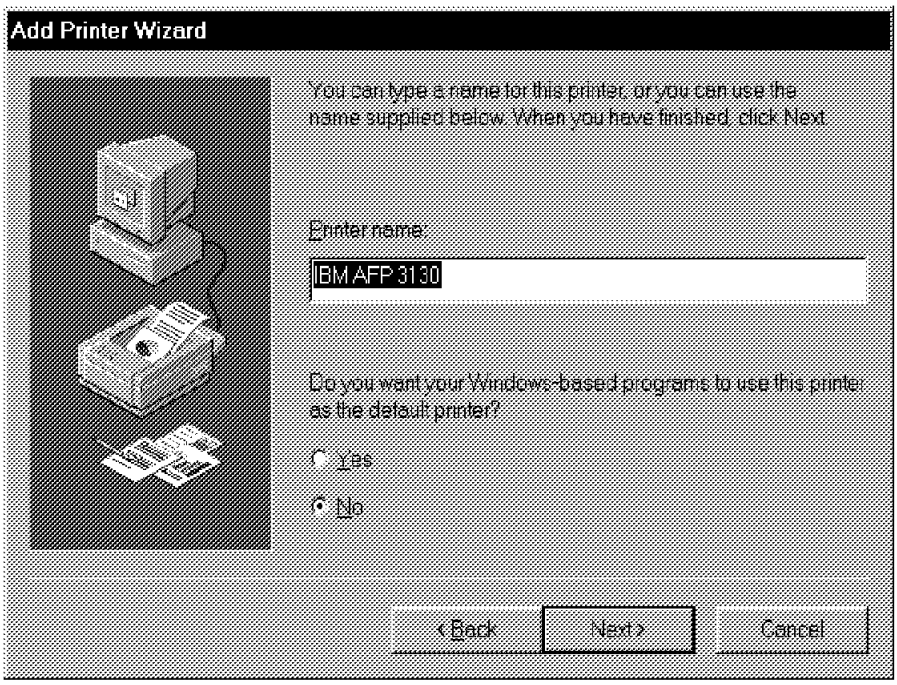


Figure 58. Windows 95 Add Printer Wizard, Select a Printer Name

8. Choose a printer name to be used to create an AFP document to file and click **Next** to get Figure 59 on page 53.

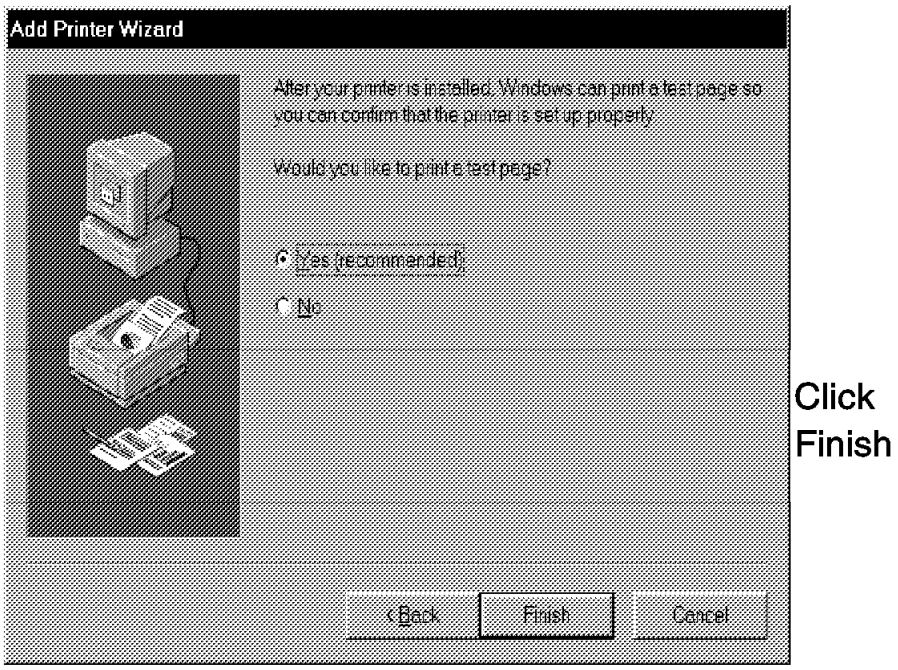


Figure 59. Windows 95 Add Printer Wizard, Click Finish

9. When you click **Finish**, a window appears that shows the loading of the printer driver, IBMAFP.DRV.

2.13.2.2 Define a Printer to Print AFP Output on Host

Follow the first five steps shown in 2.13.2.1, “Define a Printer to Create AFP Output” on page 49. Then at step 6, choose a defined printer port as shown in Figure 60 on page 54.

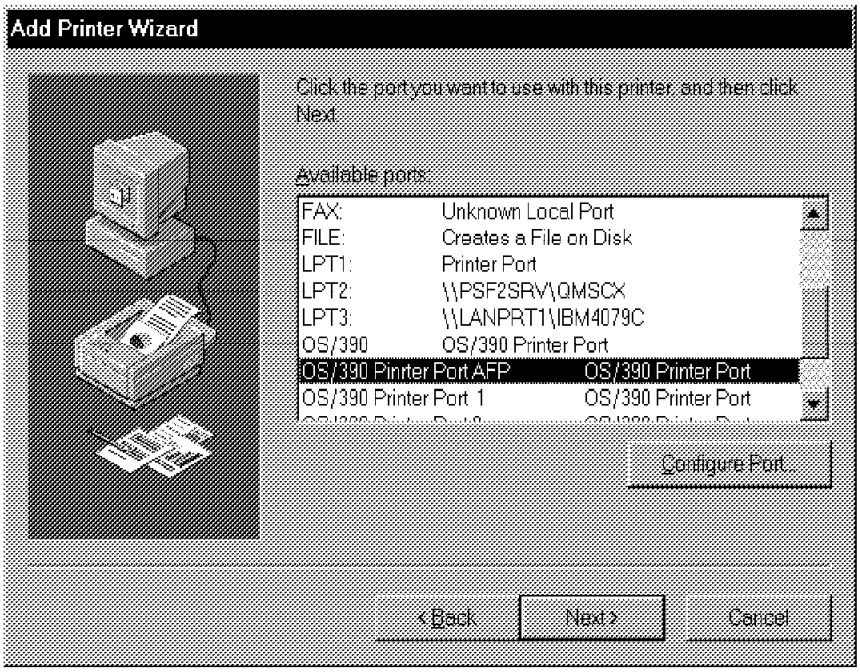


Figure 60. Windows 95 Add Printer Wizard, Choose an OS/390 Printer Port

After selecting the OS/390 port, select **Configure Port** and Figure 61 appears.

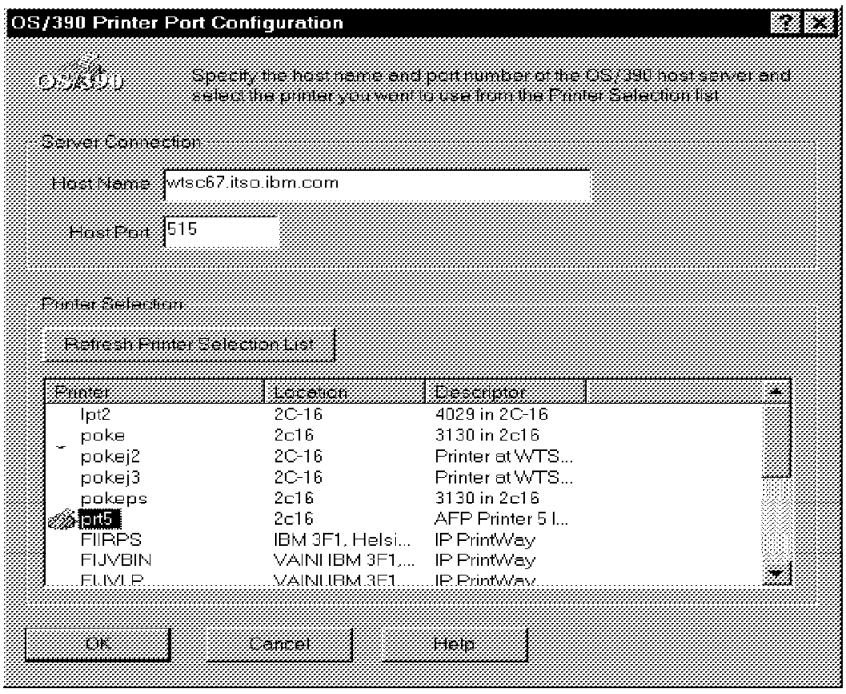


Figure 61. Windows 95 Add Printer Wizard, Select Host AFP Printer

Select a host AFP printer from the list displayed from the Print Interface table from the host, as now shown in Figure 61. When you click **OK**, Figure 62 on page 55 appears.

Figure 62. Windows 95 Add Printer Wizard, Select AFP Printer Name

As shown in Figure 62, overtype the displayed name with the AFP printer name of the new printer, prt5. When you click **OK**, the last window appears, where you click Finish. The driver is then loaded for the AFP host printer.

The AFP Printer Driver you selected completes the installation. If you choose to print a test page, you are prompted for the file name of the output file.

Back on the Printers Folder, shown in Figure 63 on page 56, if you double click the right mouse button on the driver you just installed, select **Properties**, then you can control some device capabilities and set defaults for printing options, such as paper size, orientation, and so forth.

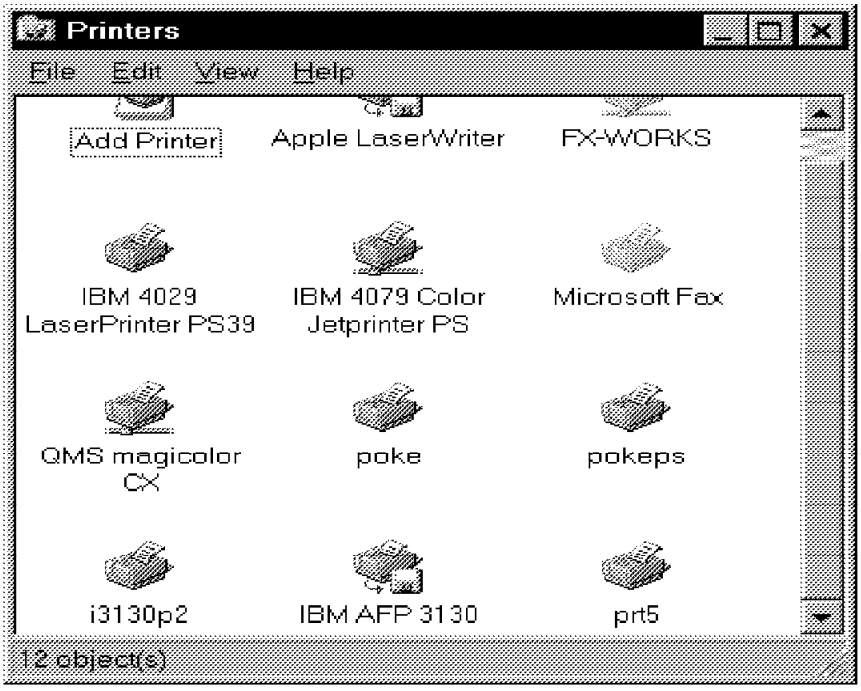


Figure 63. Windows 95 Add Printer Wizard, Defined Printers

2.13.2.3 Font Substitution Tables

A default font substitution table is shipped on the IBM AFP printer driver for Windows installation. If your font substitutions involve IBM Expanded Core Fonts other than the original IBM Core Fonts, or non-IBM fonts, the resulting documents only print at locations that have installed such fonts. It is up to the user to ensure that the fonts substituted when creating documents with the IBM AFP printer driver are actually installed on the host system(s). The fonts used in the font substitution table are usually installed if PSF is installed on the system where you print. AS/400 sites normally do not have these fonts installed.

2.13.2.4 IBM Expanded Core Fonts

The IBM Expanded Core Fonts combine the IBM Core Interchange Fonts, IBM Coordinated Fonts, and IBM BookMaster Fonts. They include the following font families:

```
Boldface
      Roman Medium
BookMaster Latin1
      Roman Medium, Roman Bold, Italic Medium, and Italic Bold
BookMaster Reverse
      Roman Medium
BookMaster Specials
      Roman Medium, Roman Bold, Italic Medium, and Italic Bold
BookMaster Specials Reverse
      Roman Medium
Courier
      Roman Medium, Roman Bold, Italic Medium, and Italic Bold
Courier APL2
      Roman Medium, Roman Bold
Gothic Katakana
      Roman Medium
```

```
Gothic Text
      Roman Medium
Helvetica
      Roman Medium, Roman Bold, Italic Medium, and Italic Bold
IBM Logo
      Roman Medium
Letter Gothic
      Roman Medium, Roman Bold
Monthob
      Roman Medium, Roman Bold, and Italic Medium
OCR-A
      Roman Medium
OCR-B
      Roman Medium
Prestige
      Roman Medium, Roman Bold, and Italic Medium
Times New Roman
      Roman Medium, Roman Bold, Italic Medium, and Italic Bold
```

The IBM Core Interchange Fonts (Courier, Helvetica, and Times New Roman) also contain a symbol collection of scientific, mathematical, and special-purpose characters in Roman Medium and Roman Bold typefaces.

The IBM Expanded Core Fonts are all derived from Type 1 font technology and are available in all five printer formats supported by AFP software, except that Monthob is available only in 240-pel bounded-box raster format. The five formats are:

- Adobe Type 1 outline format
- 240-pel unbounded-box raster format
- 240-pel bounded-box raster format
- 300-pel raster format
- AFP outline format

The IBM Expanded Core Fonts can be obtained by ordering the IBM AFP Font Collection.

2.13.2.5 Printing with an Inline Form Definition on VM

If the IBM AFP Printer Driver created an output file with an inline form definition, and your host system is VM, use the following PSF command after tagging your device and spooling your printer:

```
PSF fn ft fm (FORMDEF(F1IBMAFP FDEF38PP) cc notrc)
```

If your host system is OS/390, AS/400, or OS/2, no special options or commands are required. Print the file the same way you would any other AFP file.

2.13.2.6 Advanced Document Properties for Windows NT

An example of the Job Properties section of the driver is shown in Figure 64 on page 58. The following can be set:

- Paper size
- Orientation
- Output type:
  - Document
  - Medium overlay
  - Overlay
  - Page segment

- Inline Form definition (On or Off)
- Margin limits
- Halftone color adjustments
- Print text as graphics

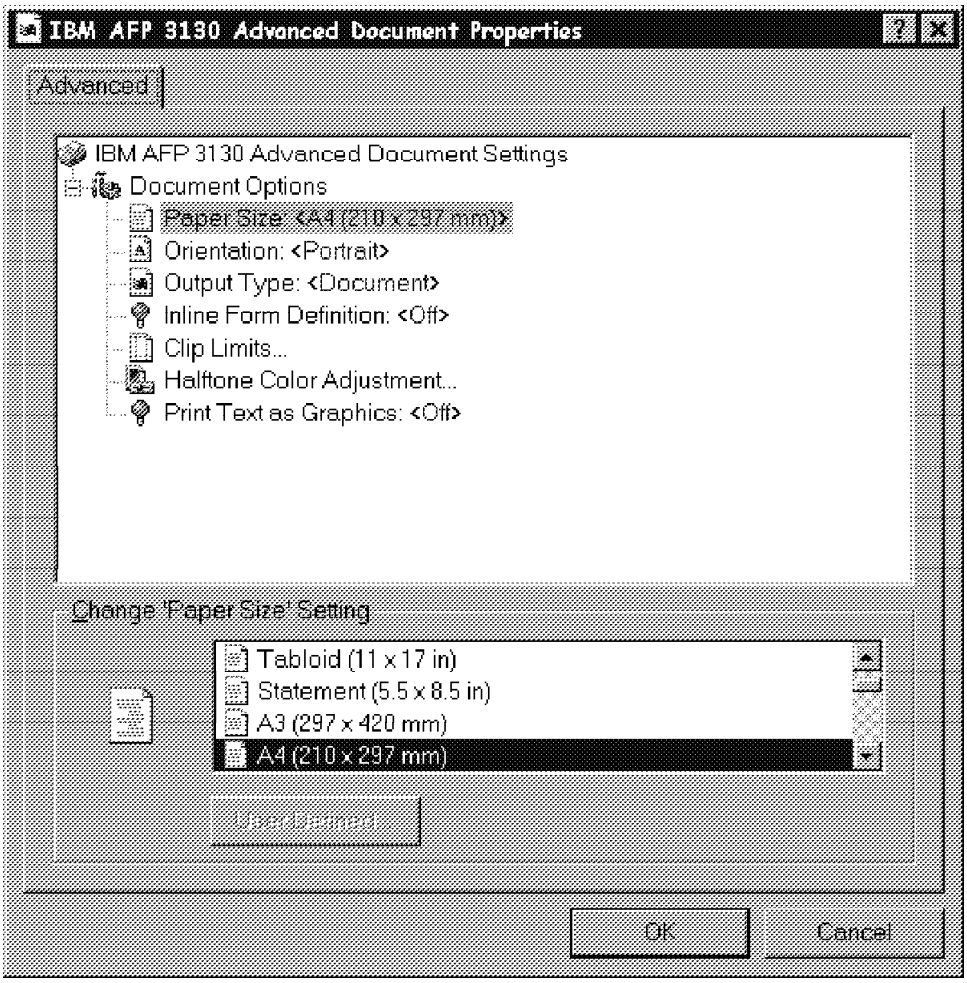


Figure 64. The Properties Section of the AFP Driver for Windows NT

2.14 IBM AFP Plug-in Viewer

The IBM AFP Plug-In Viewer allows users to view files in AFP format from an Internet browser. These documents can reside on the workstations, an attached servers or on the Internet. To view documents that reside on a S/390 host, they have to be downloaded to the workstation. Files can also be viewed that have been created by the IBM AFP Printer Driver and, if needed, routed to an OS/390 printer.

2.14.1 Installing the AFP Plug-In Viewer

To install the IBM AFP Plug-In Viewer, do the following:

1. Download the file AFPVIEWR.EXE into a temporary directory. The location of this file is specified in 2.12.1, “Printing from Windows 95 and Windows NT” on page 41.
2. Make a directory that can contain the extracted files:

```
md c:\afpvwr
```

3. Run the AFPVIEWR program to extract the files and install the viewer into the directory:

   ```
   c:\temp\afpviewr c:\afpvwr
   ```

4. Select **Next**. The Choose Destination folder is displayed indicating the default folder.
5. Select **Browse** and then enter the folder in which the viewer is to be installed. Use the c:\afpvwr folder.
6. Select **Next**. The installation of the viewer commences. The next Window to be displayed is the IBM AFP Viewer Plug-In Window.
7. Select the preferred browser if more than one browser is installed on the machine.
8. Select **Next**. The browser will now be updated with the Plug-In code.
9. When the installation completes, restart the browser if it has already been started.

2.14.2 Viewing AFP Documents

To view an AFP document, the user selects File and then Open Page from the browser and Figure 65 appears.

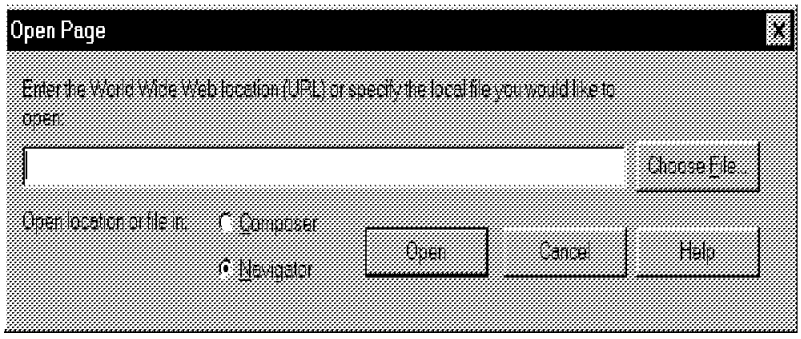


Figure 65. Open Page Window to Select Document to Be Viewed

To select an AFP document stored on your workstation, select **Choose File** and Figure 66 on page 60 appears:

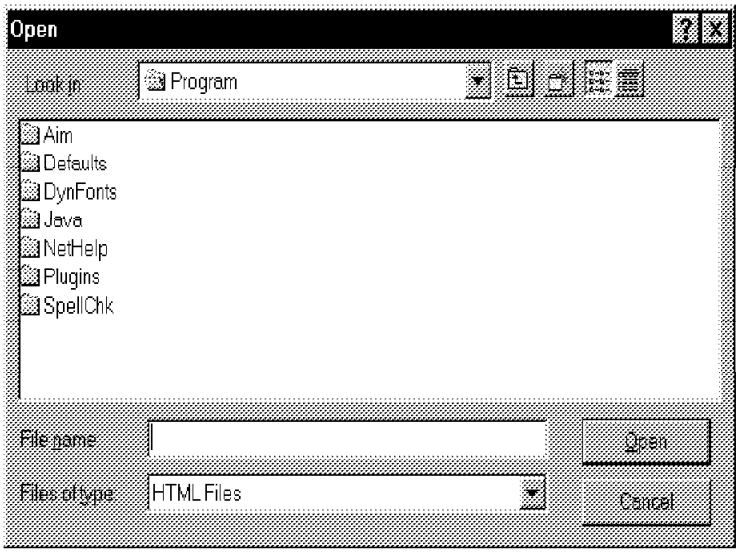


Figure 66. Window to Select AFP Document to Be Viewed

The file type must be set to *.AFP, so this must be selected as shown in Figure 67. You can type the document name in the space provided for File name or you can select a file by going to a disk where the file is stored as shown in Figure 67.

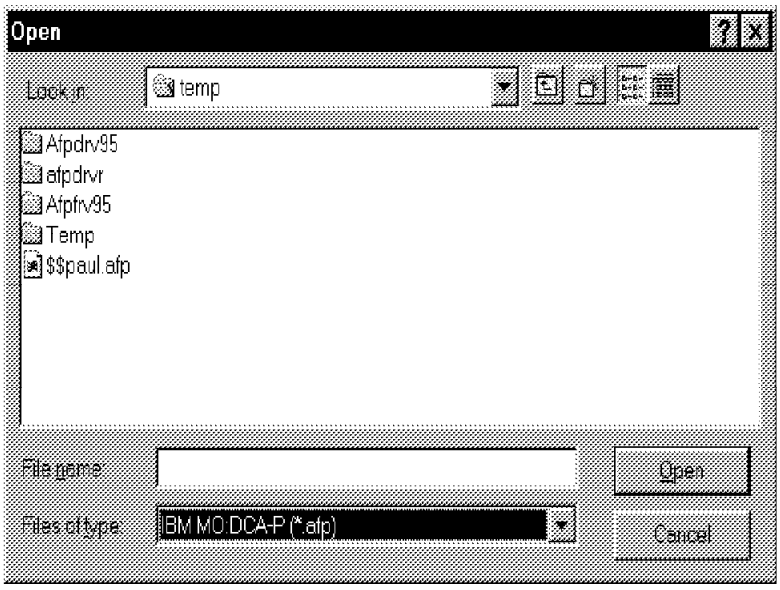


Figure 67. Window to select AFP Document to Be Viewed

After specifying the name of the AFP document on the workstation or server, Select **Open**. The browser loads the Plug-In Viewer and displays the AFP document.

The AFP Viewer (Figure 68 on page 61) allows the user to:

- Change the magnification factor with Zoom
- Rotate pages

- Magnify an area on a page
- Reset to the default view
- Toggle the display of images on or off
- Save current view settings
- Get a previously saved file
- Go to a specific page
- Find specified strings
- Find the next occurrence of a string

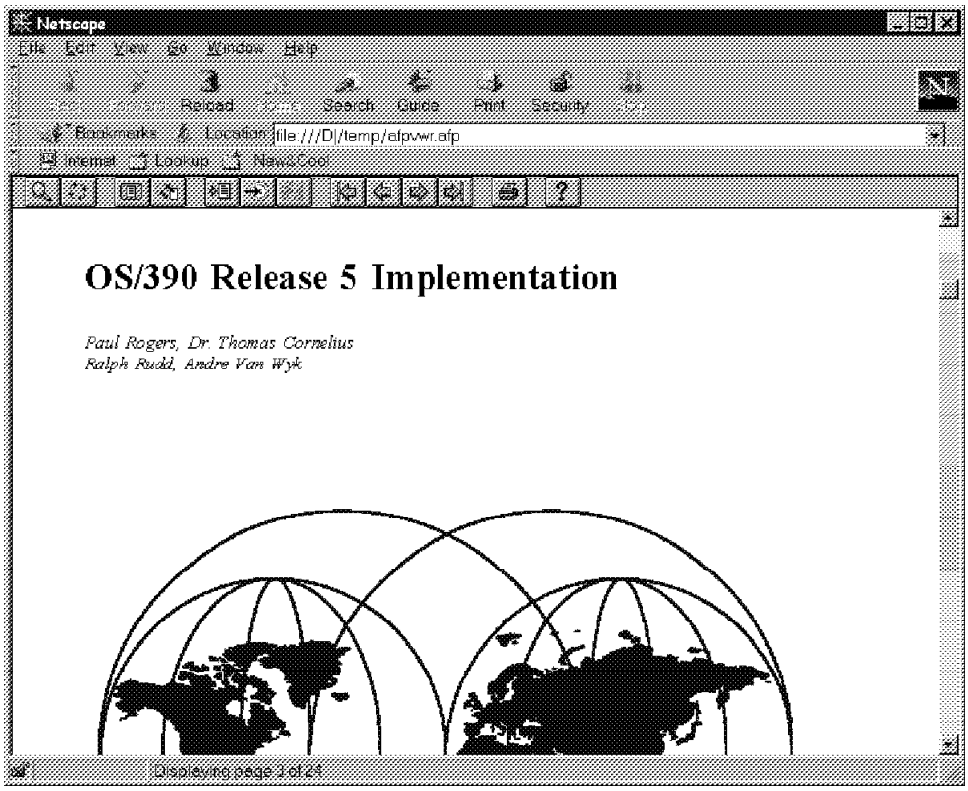


Figure 68. Document Being Viewed With the AFP Viewer

2.14.3 Printing a Document From a Browser

By selecting one of the OS/390 AFP printers from the Printer list, the user can print the selected document using the OS/390 Print Interface.

2.15 Print Server Maintenance

For toleration support of the new JCL keywords in a JES3 environment, apply the PTF associated with APAR OW21512.

For JES3 support of new JCL keywords used by IP PrintWay, apply the PTF associated with APAR OW30924.

Note: This APAR applies only to the OS/390 R5 level of JES3.

2.16 JES2 Release 5

OS/390 Release 5 JES2 provides the interface by which the OS/390 Printer Server prints SYSOUT data sets for clients and allows them to obtain status for, view, cancel, or release of SYSOUT data sets previously submitted.

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF OUTPUT ALL CLASSES ALL FORMS     LINES 3           DATA SET DISPLAYED
COMMAND INPUT ===>                                        SCROLL ===> PAGE
NP   JOBNAME  JOBID    OWNER    PRTY C FORMS    DEST                 TOT-REC
     ROGERS4  STC13167 ROGERS    144 J STD      <IP>                       1
     ROGERS   PSO00032 STC       144 J STD      <IP>                       1
     ROGERS   PSO00034 STC       144 J STD      <IP>                       1
```

2.16.1 JES2 Maintenance

The Print Server function is in the base of OS/390 Release 5. No special installation procedures or initialization parameters are required.

On lower releases of JES2, APAR OW29209 is required if those releases are sharing the spool in a sysplex environment and coexist with a OS/390 Release 5 JES2.

2.16.2 Print Server Maintenance

For JCL keyword support for specifying IP address on the DEST parameter in a JES2 environment, apply the PTFs associated with the following APARs:

- OW21839, OW21918, and OW21924

2.17 JES3 Release 5

JES3 has changed to support allocation of and access to SYSOUT data sets by the OS/390 Print Server on behalf of clients. During dynamic allocation of a SYSOUT data set, the OS/390 Print Server requests JES3 to assign a client token (CTOKEN) to the data set. OS/390 Print Server later uses this token as a selection filter when it performs a subsystem interface (SSI) call to SAPI or Extended Status.

2.18 JES3 Release 5 and IP PrintWay

JES3 output services are enhanced to support IP Printway, a feature of OS/390 Version 1 Release 3, by allowing specification of an IP address on the DEST= keyword in the //OUTPUT JCL statement and the OUTDES statement. JES3 support for the IP address includes FSS and SYSOUT application program interface support, command changes, networking support, output service writer and PSO implications.

Chapter 3. Accessing ISPF from the World Wide Web

OS/390 Release 5 includes a new feature called ISPF Application Server. This feature allows an ISPF application to be accessed from the World Wide Web. In this chapter, the functionality of this process and the installation procedure is discussed, and some detailed examples of establishing connections between a Web browser and TSO/ISPF are shown.

3.1 How the ISPF Application Server Works

The following software components participate in the connection between a Web browser and TSO/ISPF (see Figure 69):

- A Web browser running on a workstation.
- A Web server and the ISPF Application Server running on the same Internet address.
- TSO/ISPF and JES running on an OS/390 System.

The capabilities of a Web server, which are necessary to connect a browser to ISPF, are integrated in the ISPF Application Server. It is sufficient to install the latter one on the server.

The Web browser and the Application Server may run on the same workstation, which may be advantageous for test purposes.

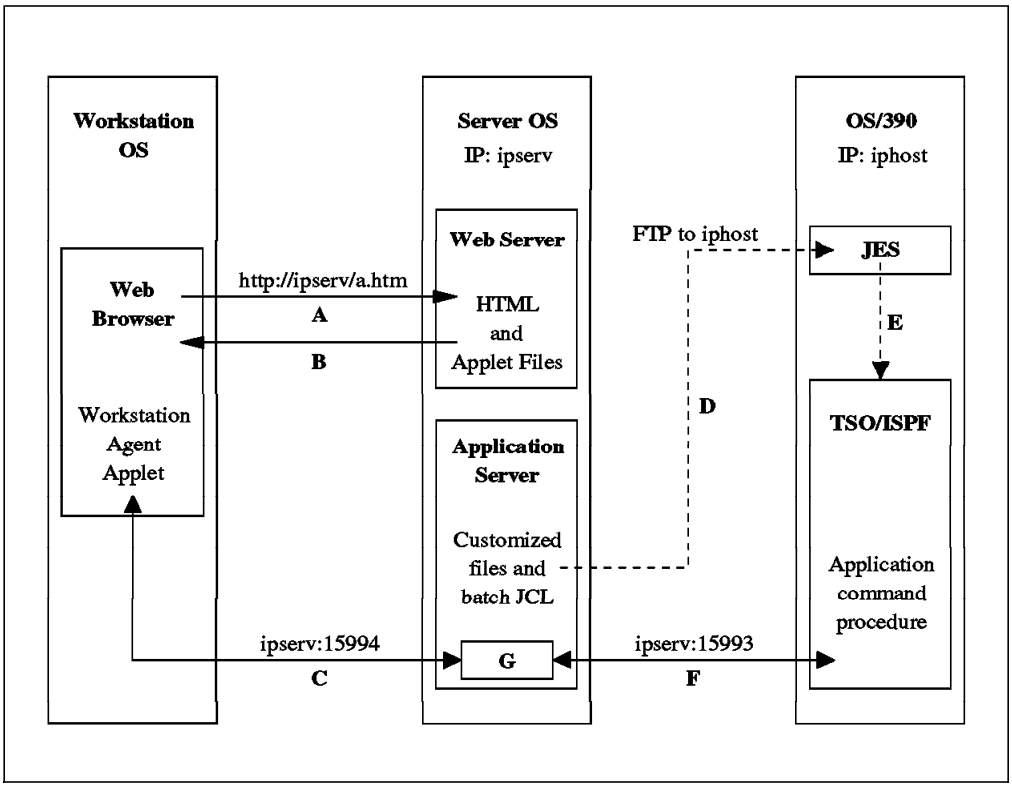


Figure 69. Functionality of the ISPF Application Web Server

To establish a connection between a Web browser and a TSO/ISPF application on the host, the following steps are performed (see Figure 69):

1. A Uniform Resource Locator (URL) for an ISPF Workstation Agent Applet HTML is selected from the Web browser. The URL consists of the IP address of the server (ipserv) and the name of the HTML file for the chosen application (a.htm) (**A**).
2. A connection to the Web server is established and the requested HTML and its associated Java applet are sent to the browser (**B**).
3. The applet is interpreted by the Java virtual machine of the Web browser. It connects the Web browser to the Application Server using the IP address of the server and the Workstation port number (default is 15994) (**C**).
4. In the OS/390 system, there has to be an address space communicating with the Application Server. Depending on the chosen application, this may be an existing TSO/ISPF session or batch job. This type of session is called *user-initiated*.

   Another option is to let the Application Server transfer JCL via FTP to the Host (**D**) and submit it to JES to create a new address space (**E**). This is called a *server-initiated* session.
5. In both cases mentioned in point 4, an appropriate command supplied in this host address space builds up the connection to the Application Server by specifying the application name, the IP address of the Application Server, and the Application port number (default is 15993) (**F**).
6. The Application Server matches the two connections and passes the data between the Workstation Agent Applet and the TSO/ISPF application (**G**).

3.2 Advantages of the ISPF Application Server

After successful execution of these steps, one gets an ISPF application running on the OS/390 system, with which the user interacts via a GUI interface on a workstation. This seems to be similar to the use of the ISPF Workstation Agent, which was introduced with ISPF Version 4. But there are two distinct differences between using the ISPF Application Server and the ISPF Workstation Agent:

- Using the Application Server, the user does not need to take care about upgrading the level of the ISPF workstation software. He always receives the current level of the applet from the Application Server by standard Internet techniques.
- Using the Workstation Agent, only a user-initiated session is supported: The address space, in which the application runs, has to be created by a TSO logon or by somebody submitting a batch job. In contrast to this, in a server-initiated session, the Application Server creates an address space on the host system.

3.3 Installing the ISPF Application Server

In this section we discuss the steps which must be executed to install the Application Server.

First we check the software prerequisites and give hints regarding how they can be fulfilled if a Windows 95 or Windows NT system is used for the Application Server and the Web browser. Then we show how to use the Application Server installation utility. Finally we make some remarks about the Application Server files, which are created on the server system by the installation procedure.

3.3.1 Software Prerequisites

The following software prerequisites must be satisfied in order to install and use the ISPF Application Server:

1. OS/390 Version 2 Release 5 is the minimum level of the operating system installed on the host.
2. TCP/IP connectivity between all participating systems must be enabled.
3. The Java Runtime Environment (JRE) at level 1.1.1 must be available on the Server workstation.
4. The Web browser used to invoke the ISPF Workstation Agent Applet must support JRE at level 1.1.1.

   If a Web browser, for example, Microsoft′s Internet Explorer 4.0, does not support JRE at level 1.1.1, Sun′s Java Plug-In software (formerly code-named “Java Activator”) may be used.

We installed the ISPF Application Server on a Windows 95 and a Windows NT workstation. The corresponding JRE can be downloaded from the Sun Java Home Page in the World Wide Web. One can either use the Java Development Kit (JDK) or the Java Runtime Environment (JRE), which is a subset of the JDK. We installed JDK and JRE at the level 1.1.5. To install the JDK or JRE follow, the instructions on the Web page.

Under Windows 95 and Windows NT we tested two Web browsers, which support the required JRE level:

- Netscape Communicator 4.04

  It is available on the Netscape Home Page. We had to add the JDK support in an additional “Smart Update”step.

- HotJava Browser 1.1.2

  This browser can be downloaded from the Sun Java Home Page.

We also used Microsoft′s Internet Explorer 4.0 in combination with Sun′s Java Activator Early Access Release 3 (now named “Java Plug-In”) to invoke the ISPF Workstation Agent Applet. The method of downloading the Java Activator software to your browser workstation is described in 3.5.3, “Using Sun′s Java Virtual Machine on Win32 Platforms” on page 83.

3.3.2 The ISPF Application Server Installation Utility

After satisfying all software requirements, the following steps must be executed to install the ISPF Application Server on a Windows 95/NT workstation:

1. Download the ISPF Application Server installation routine to the workstation.

   The installation routine resides in the member ISPZ001 in the data set hlq.SISPJSRV, where hlq are the high level qualifiers of the ISPF target libraries on the OS/390 system. To download it to the file install.zip, use FTP (see Figure 70 on page 66). Ask your network administrator for the IP address of your host.

```
C:\>ftp iphost
Connected to iphost.
220-FTPD1 IBM FTP CS/390 Release 5 at XXXXXXX, 21:34:20 on
1998-03-16.  220 Connection will close if idle for more than 5 minutes.
User (iphost:(none)): userid
331 Send password please.
Password:
230 USERID is logged on.  Working directory is "USERID.".
ftp> cd ..
250 "" is the working directory name prefix.
ftp> cd hlq.sispjsrv
250 "HLQ.SISPJSRV" partitioned data set is working directory
ftp> bin
200 Representation type is Image
ftp> get ispz0001 install.zip
200 Port request OK.
125 Sending data set HLQ.SISPJSRV(ISPZ0001)
250 Transfer completed successfully.
79237 bytes received in 0.44 seconds (180.08 Kbytes/sec)
ftp> quit
221 Quit command received. Goodbye.
```

Figure 70. Download of the Installation Utility

2. Start the installation utility as a Java application.

   On a Windows 95/NT system, this is done by invoking either the Java Development Kit java command or the Java Runtime Environment jre command from a DOS command prompt:

   ```
   C:\>java -classpath install.zip;%classpath% Install
   ```

   or

   ```
   C:\>jre -cp install.zip; Install
   ```

   For the first command, the current classpath indicated by %classpath% must be set according to the instructions on the JDK download Web page. The name of the installation utility Install is case-sensitive in both commands.

3. Go through the installation utility.

   a. In the first window, press the **Accept** pushbutton.

   b. In the next window, shown in Figure 71 on page 67, enter the name of your hlq.SISPJSRV host data set and the target directory on the workstation, for example ispf. If you do not enter the absolute path name, the target directory will be created as subdirectory of the current directory. Select the **Next** pushbutton.

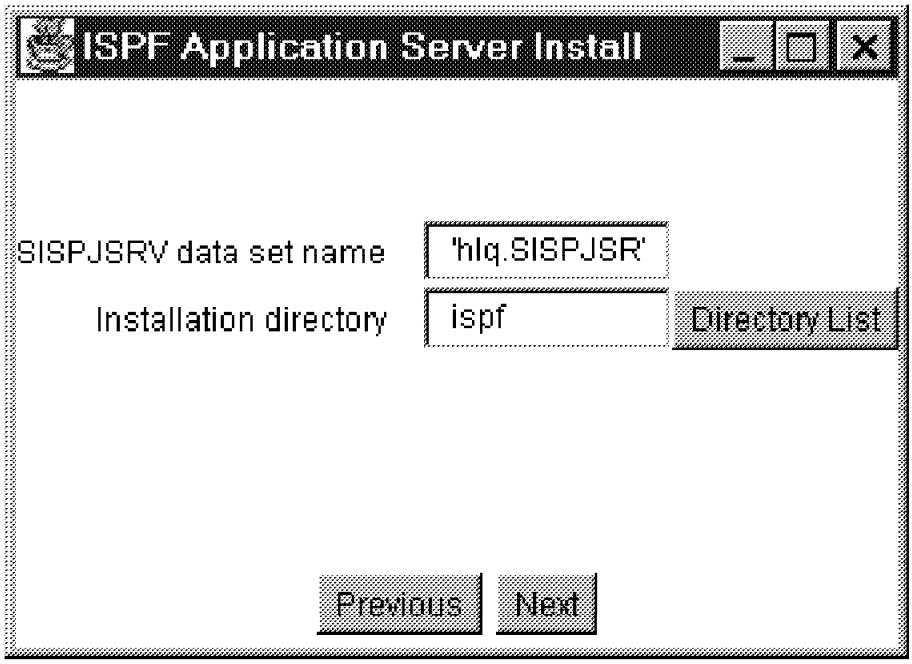


Figure 71. ISPF Application Server Installation Utility - Dialog 1

c. On the third dialog panel (see Figure 72), fill in the IP address of the OS/390 system (iphost in Figure 69 on page 63) and your user ID and password. After choosing the **Install** pushbutton, the utility starts to install the ISPF Application Server files into the specified target directory ispf.

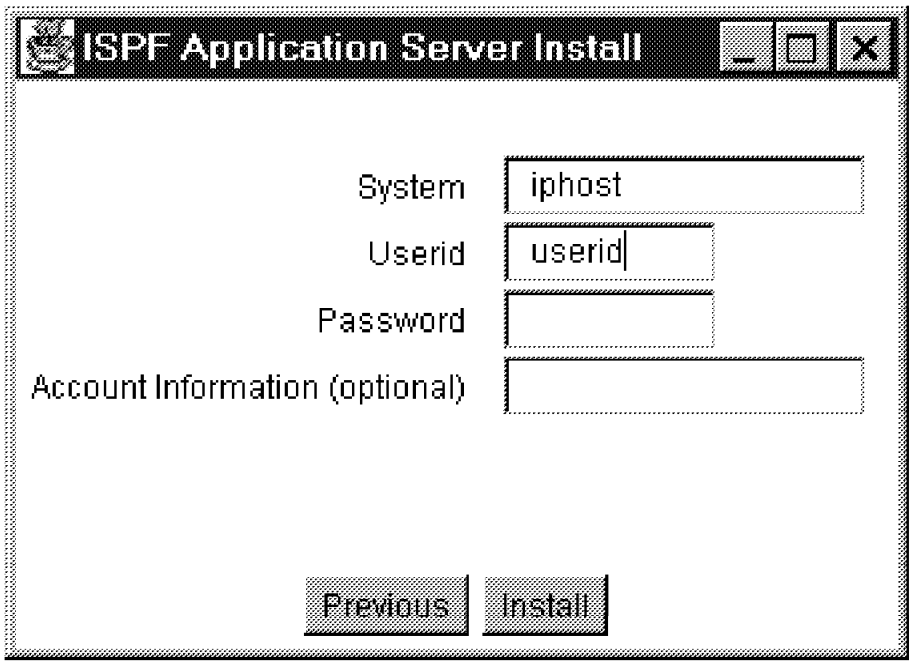


Figure 72. ISPF Application Server Installation Utility - Dialog 2

d. Finally, quit the installation dialog by pressing the **OK** pushbutton in the last window.

3.3.3 The ISPF Application Server Files

The directory structure of the ISPF Application Server is shown in Figure 73.

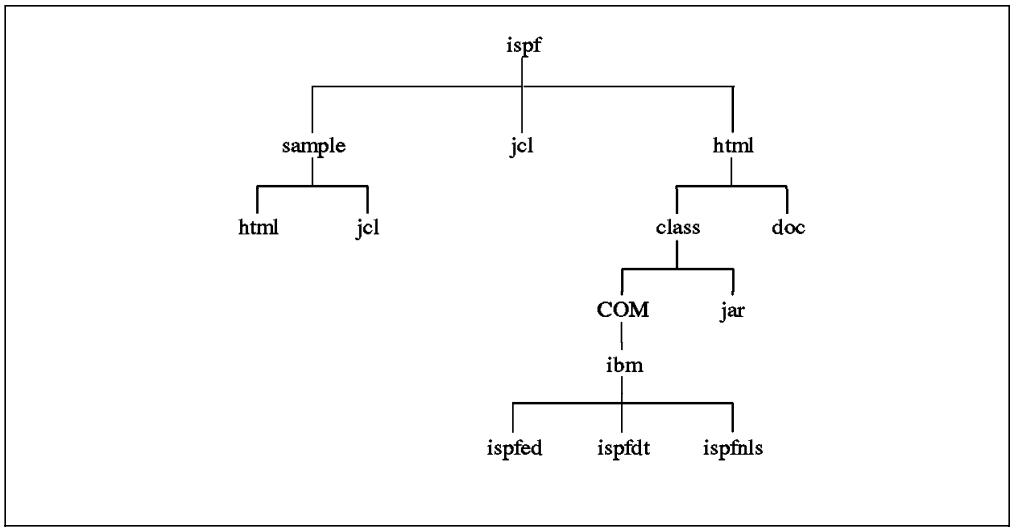


Figure 73. ISPF Application Server Dirctory Structure

The installation utility copies files from the \ispf\sample\ directories to the corresponding \ispf directories (for example, from \ispf\sample\html to \ispf\html), provided that the individual files do not already exist in the target libraries. Therefore, the installation routine can be rerun without replacing changes that the Application Server administrator may have made to the sample files.

A documentation of the ISPF Application Server in HTML format is contained in the \ispf\html\doc directory. To read it, invoke the \ispf\html\doc\contents.htm file from a Web browser.

The Workstation Agent Applet class files are provided in two formats:

- The class files, which are consolidated into Java archive (JAR) files, reside in the ispf\html\class\jar\ directory.
- The individual class files are in the ispf\html\class and ispf\html\class\COM\... directories.

For better performance, it is advantageous to use the JAR files, if the are supported by the Web browser. To allow Web browsers to use both types of class files, the following CODEBASE and ARCHIVE definition should be included in the APPLET tag of the Workstation Agent Applet HTML for an application:

```
<APPLET CODE="wsb.class"
ARCHIVE="jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar"
CODEBASE="http://ipserv/class" height="570" width="760">
```

We discuss the Workstation Agent Applet HTML in more detail in 3.5.1.1, “The HTML for a User-Initiated Session” on page 74 and 3.5.2.2, “The HTML for a Server-Initiated Session” on page 80.

If a Web server different from the one included in the ISPF Application Server is used, the Workstation Agent Applet HTML and applet files must be copied from the installation directories to the corresponding directories of the alternative Web server. In this case, the CODEBASE statement given in the previous

example has to be changed to reflect the new location of the classes files. For example, if the applet files are in the subdirectory \classes\ispf\ of the default HTML directory of the Web server, then the APPLET tag reads like the following one:

```
<APPLET CODE="wsb.class"
ARCHIVE="jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar"
CODEBASE="http://ipserv/classes/ispf/ height="570" width="760">
```

3.4 ISPF Application Server Basic Configuration

Now we are ready to start the ISPF Application Server and to perform two important configuration steps: Starting the integrated Web server and defining the port numbers that the Application Server will use.

3.4.1 Starting the ISPF Application Server

To start the ISPF Application Server, change to the Application Server installation directory (ispf in Figure 71 on page 67). Then enter one of the following commands (depending on what you want to use JDK or JRE), in a DOS command prompt:

```
java -classpath server.zip;.;html\class;%CLASSPATH% ApplicationServer
```

or

```
jre -cp server.zip;.;html\class; ApplicationServer
```

As mentioned in 3.3.2, “The ISPF Application Server Installation Utility” on page 65, %classpath% in the first statement has to be defined as described on the JDK download Web page.

After activating the Application Server, the window shown in Figure 74 on page 70 appears on the screen.

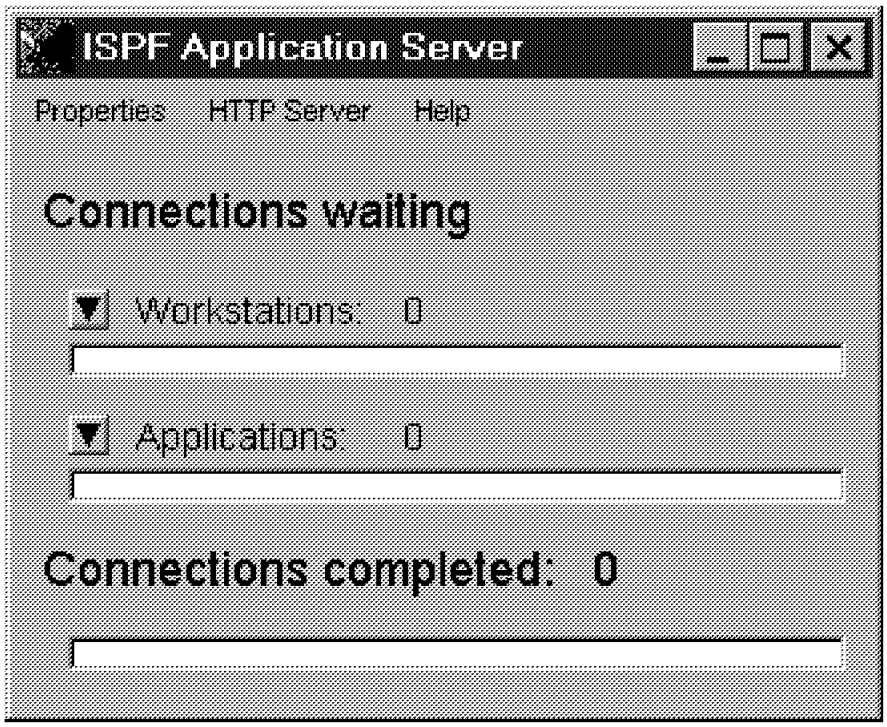


Figure 74. ISPF Application Server

3.4.2 Starting the Integrated Web Server

If you want to use the Web server integrated in the Application Server instead of an external one, you have to start it. This is done by choosing the **HTTP Server** pull-down menu (see Figure 74) and selecting the **Start** option (see Figure 75 on page 71).

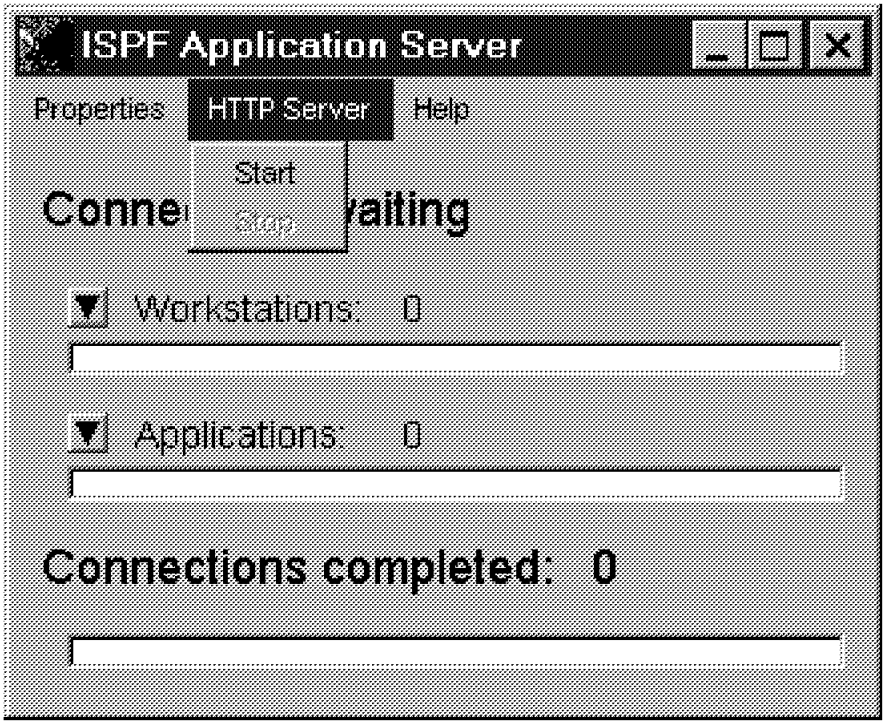


Figure 75. Starting the Web Server Facility

If the integrated Web server should be activated automatically every time the Application Server is started, the Enable at startup check box in the window shown in Figure 76 on page 72 has to be selected. You reach this panel from the one shown in Figure 74 on page 70 by selecting **Properties**, then **General...** and then **HTTP Server**.

If you change the port number of the integrated Web server from its default value 80 (see Figure 76 on page 72), you must add the new port number to the URL of your Workstation Agent Applet HTML, for example in Figure 69 on page 63, step **A**:

```
http://ipserv:new_port_number/a.htm
```

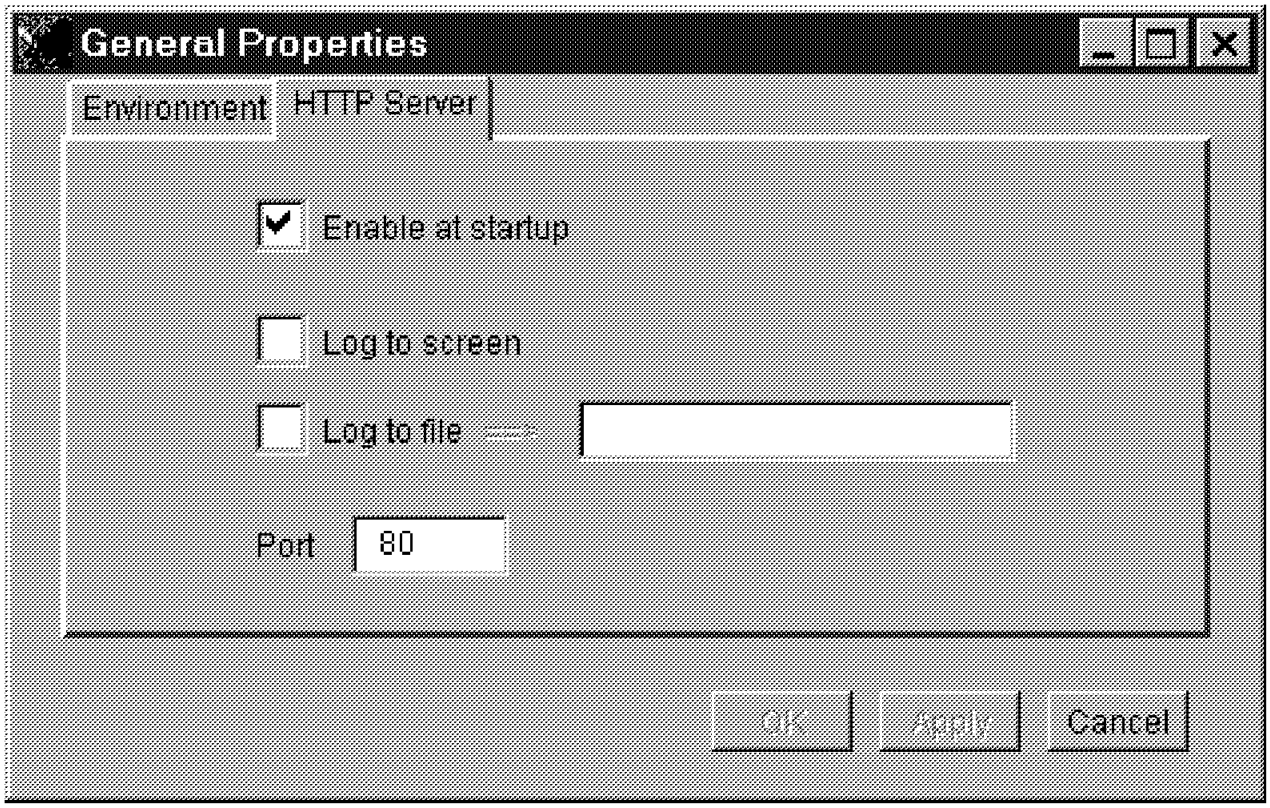


Figure 76. HTTP Server Properties

3.4.3 Defining the Application and Workstation Port Numbers

In the Environment panel (see Figure 77 on page 73), which can also be accessed from the General... choice of the Properties pull-down menu in the Application Server main window (Figure 74 on page 70), you may change the Application and Workstation port numbers. The port numbers play an important role in establishing the connection between the Web browser and TSO/ISPF session on the host, as we discussed in 3.1, “How the ISPF Application Server Works” on page 63.

In Figure 77 on page 73, both port numbers equal their default values. It may be necessary to change the Application port number to 15995, for example, if you have installed the ISPF Workstation Agent (WSA) on the server workstation and you connect it via TCP/IP to an ISPF host session, because the default port number of the WSA is also 15993.

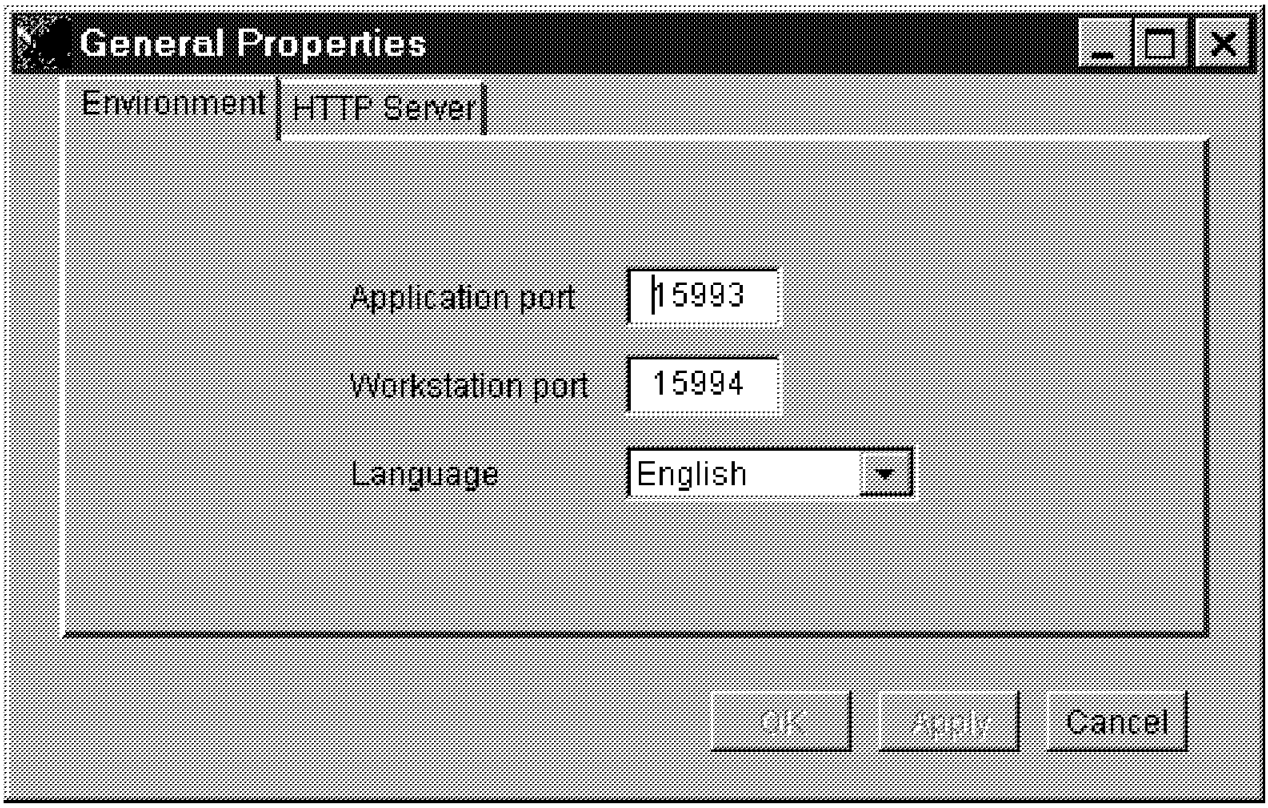


Figure 77. Environment Properties

3.5 Connection Examples

After installing the ISPF Application Server on the server workstation, there remain three major steps to be executed to establish a connection between a Web browser and an ISPF address space on a host:

- The Application Server has to be configured by creating the ports to be used.
- An ISPF Workstation Agent Applet HTML must be coded as shown in Figure 78 on page 74.
- The OS/390 system has to be have a TSO/ISPF address space. There are two ways to initiate a TSO/ISPF session on the host:
  - User-initiated session by submitting a batch job or by a logon to TSO
  - Server-initiated session where the Application Server submits the JCL to the host, as shown in Figure 69 on page 63.

3.5.1 Connecting via a User-Initiated Session

If you choose to establish the TSO/ISPF session on the host by a user-initiated method, in Figure 69 on page 63, the steps **D** and **E** are not executed. The first of the two options shown in point 4 on page 64 is implemented.

To start the Application Server, execute the following steps which are described in detail in 3.4, “ISPF Application Server Basic Configuration” on page 69:

1. Issue the following command from your MS-DOS session to start the Application Server installation by using the ISPF directory:

```
java -classpath server.zip;.;html\class;%CLASSPATH% ApplicationServer
```

or

```
jre -cp server.zip;.;html\class; ApplicationServer
```

2. Start the Web server facility.[1]
3. Note the Workstation and Application port numbers (see Figure 77 on page 73).

You will need the port numbers when you code the Workstation Agent Applet HTML for this example and when you execute the command, which builds the connection to the Application Server, in the host address space.

3.5.1.1 The HTML for a User-Initiated Session

If you use a Web browser, which supports the required JDK or JRE level (see 3.3.1, “Software Prerequisites” on page 65), code the following ISPF Workstation Agent Applet HTML and save it with the name interact.htm in the HTML directory of your Web server (ispf\html\ in Figure 73 on page 68).

Note: If you are using Microsoft’s Internet Explorer, the HTML has to be modified as described in 3.5.3, “Using Sun’s Java Virtual Machine on Win32 Platforms” on page 83.

```
<HTML>
<TITLE> ISPF Workstation Agent Applet</TITLE>
<BODY>
<APPLET CODE="wsb.class"
ARCHIVE="jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar"
CODEBASE="http://ipserv/class" height="570" width="760">
<PARAM NAME=APPLICATION VALUE=>
<PARAM NAME=AUTOCONNECT VALUE=n>
<PARAM NAME=BATCH VALUE=n>
<PARAM NAME=COMMAND VALUE=>
<PARAM NAME=MAXWAIT VALUE=30>
<PARAM NAME=PANELSINBROWSER VALUE=n>
<PARAM NAME=PASSWORD VALUE=>
<PARAM NAME=PORT VALUE=15994>
<PARAM NAME=PROCEDURE VALUE=>
<PARAM NAME=REQAPPLICATION VALUE=y>
<PARAM NAME=REQCOMMAND VALUE=n>
<PARAM NAME=REQPASSWORD VALUE=n>
<PARAM NAME=REQPROCEDURE VALUE=n>
<PARAM NAME=REQSYSTEM VALUE=n>
<PARAM NAME=REQUSER VALUE=n>
<PARAM NAME=RUNBUTTONTEXT VALUE=>
<PARAM NAME=SHOWOPTIONS VALUE=y>
<PARAM NAME=SYSTEM VALUE=>
<PARAM NAME=USER VALUE=>
</PARAM>
</APPLET>
</BODY>
</HTML>
```

Figure 78. HTML for User-Initiated Session

[1] As an example of an external Web server we also used Microsoft’s Personal Web Server under Windows 95.

In Figure 78, a description of the bold-typed parameters in the HTML is as follows:

CODEBASE The CODEBASE parameter describes the URL to be entered by the user in his browser.

PORT This parameter must be set to the workstation port number, which was defined in Figure 77 on page 73.

REQUSER The value of REQUSER indicates that no user entry is required, because the user connects to an already existing TSO/IPSF session.

REQPASSWORD The value of REQPASSWORD indicates that no user entry is required, because the user connects to an already existing TSO/IPSF session.

Other parameters are discussed in 3.5.2.2, “The HTML for a Server-Initiated Session” on page 80. A detailed description of the parameters can be found in *OS/390 V2R5.0 ISPF Application Server Guide and Reference*, SC34-4619.

3.5.1.2 Connecting the Host System to the Application Server

On the host side there are two types of address spaces which can connect to the Application Server in a user-initiated session. The two types are:

- An interactive TSO/ISPF session
- A TSO/ISPF batch job

Connecting an Interactive TSO/ISPF Session

1. Logon to TSO/ISPF with your user ID.
2. Invoke the Initiate Workstation Connection panel, which can be found under the Workstation pull-down menu in the Settings panel (ISPF Option 0).
3. Complete the panel as shown in Figure 79 on page 76.
   - ipserv has to be substituted with the IP address of the server workstation, which you will get from your network administrator.
   - 15993 is the application port number defined in Figure 77 on page 73 (see also Figure 69 on page 63, step **F**).
   - The application name is example1.
   - The host session will wait for a maximum of 360 seconds for the connection to the applet running in the Web browser.

```
                    Initiate Workstation Connection
Command ===>

Workstation Connection                    GUI Network Protocol
3  1. With GUI display                    1  1. TCP/IP
   2. Without GUI display                    2. APPC
   3. Connect to ISPF Application Server     3. Use ISPDTPRF file

GUI Title

TCP/IP Address
ipserv:15993
APPC Address

Application Name
example1

Maximum Connection Wait Time . . . 360

Host Codepage  . . . 37       Host Character Set . . . 697

GUI Window Frame              Default Window Background Color
1  1. Standard (STD)          1  1. Dialog (DLG)
   2. Fixed (FIX)                2. Standard (STD)
   3. Dialog (DLG)

Press ENTER to initiate a session.  Press EXIT or CANCEL to return
without initiating a session.
```

Figure 79. Initiate Workstation Connection

4. Press the Enter key.

   You receive the following message:

```
You have requested to connect to the ISPF application server. You must
enter the following application name in your workstation agent applet in
order to run your ISPF application: example1-USERID.
Press Enter to submit
your request to connect to the ISPF application server.  Note that any
additional updates you make to this panel before hitting enter are
ignored.
```

5. Press Enter again.

The connection to the Web server is established (step **F** in Figure 69 on page 63) and an application waiting for a connection is listed in the Application Server main window (see Figure 82 on page 78).

Connecting a Batch TSO/ISPF Session

1. Type in a procedure JCL like the following one and save it with the name TSOISPF to a library in the PROCLIB concatenation.

```
//TSOISPF  PROC ISPFUSR=
//*
//COPYPROF EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DSN=&ISPFUSR..profile,DISP=SHR
//SYSUT2   DD DSN=&&PROFILE,UNIT=VIO,DISP=(NEW,PASS),
//            SPACE=(TRK,(1,1,5)),DCB=(LRECL=80,DSORG=PO,RECFM=FB)
//*
//TSO      EXEC PGM=IKJEFT01,TIME=1440,REGION=4M,DYNAMNBR=75
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//ISPPROF  DD DSN=&&PROFILE,DISP=(OLD,DELETE)
//ISPMLIB  DD DSN=hlq.SISPMENU,DISP=SHR
//ISPPLIB  DD DSN=hlq.SISPPENU,DISP=SHR
//ISPSLIB  DD DSN=hlq.SISPSENU,DISP=SHR
//ISPTLIB  DD DSN=hlq.SISPTENU,DISP=SHR
//SYSPROC  DD DSN=hlq.SISPCLIB,DISP=SHR
//         PEND
```

Figure 80. TSO/ISPF Procedure

In the first step of the procedure the ISPF profile data set of the user is copied, so that the user can run more than one job concurrently.

Change the procedure where necessary to satisfy your installation parameters.

2. Submit a batch job like the one shown in Figure 81 on the OS/390 system.

```
//xxxxxxxx JOB ACCTN#,userid,NOTIFY=userid,MSGLEVEL=(1,1)
//STEP1    EXEC TSOISPF,ISPFUSR=userid
//TSO.SYSTSIN DD *
  ISPSTART PANEL(ISR@PRIM) NEWAPPL(ISR) GUI(ipserv:15993) +
        GUIWEB(EXAMPLE1,,360)
```

Figure 81. TSO/ISPF Batch Job

The definition of the boldtype parameters is the same as described in "Connecting an Interactive TSO/ISPF Session" on page 75.

After the connection is established (step **F** in Figure 69 on page 63), this is indicated on the ISPF Application Server window, where an application waiting for a connection is listed (see Figure 82 on page 78).

Figure 82. Waiting Application Connection

3.5.1.3 Connecting the Workstation to the Application Server

To be able to connect the workstation to the Application Server, the HTML of the Workstation Agent Applet, coded in 3.5.1.1, "The HTML for a User-Initiated Session" on page 74, has to be invoked from the Web browser:

1. Enter the following Uniform Resource Locator on the Web browser (see Figure 69 on page 63, step **A**): The Workstation Agent Applet is loaded onto the browser workstation (step **B**) and starts running.

   `http://ipserv/interact.htm`

2. Enter example1-USERID as the application name on the logon window, which is displayed by the Workstation Agent Applet.
3. Select the **Contact** pushbutton.

The connection from the Applet to the Application Server will now be established (step **C**). The Application Server identifies this new connection by its name and matches it to the waiting application connection with the corresponding name (step **G**).

The GUI session starts on the browser workstation and the ISPF Application Server control window lists one completed connection with the name EXAMPLE1-USERID TOKEN=<none>.

To quit the connection between the applet and the TSO/ISPF on the host, exit the GUI session by pressing the F3 key several times. The interactive TSO/ISPF host session then can be continued from the READY mode. The TSO/ISPF batch job terminates.

3.5.2 Connecting to a Server-Initiated Session

In this section we discuss the second option mentioned in point 4 on page 64: The Application Server transfers JCL via TCP/IP to the host and submits it to JES (step **D** and **E** in Figure 69 on page 63).

3.5.2.1 Configuring the ISPF AS for a Server-Initiated Session

Start the Application Server in the same way as described in 3.4, "ISPF Application Server Basic Configuration" on page 69. Execute the following steps:

1. Change to the Application Server installation directory and enter one of the following commands:

   ```
   java -classpath server.zip;.;html\class;%CLASSPATH% ApplicationServer
   ```

   or

   ```
   jre -cp server.zip;.;html\class; ApplicationServer
   ```

2. Start the Web server facility as indicated in Figure 75 on page 71 or Figure 76 on page 72.
3. Note the Workstation and Application port numbers, which can be defined on the window shown in Figure 77 on page 73.

In addition to the previous example in this case, it is necessary to define the ISPF application, which should be started on the OS/390 system, to the Application Server:

1. Select Application... from the Properties pull-down menu of the Application Server main window (Figure 74 on page 70)
2. Click on the **Add** button.
3. Define the properties of the application (see Figure 84 on page 80):
   a. As a name of the application you may choose the name Example2.
   b. According to this, the name of the JCL file should be Example2.jcl.
   c. Under System, enter the TCP/IP name of the system on which the ISPF application executes. As shown in Figure 69 on page 63, this is the iphost. Ask your network administrator for the IP address of your host.
   d. Leave the User Id and Password fields blank, since in this example we code the Application Agent HTML in that way, that the user is prompted for these values after the applet has started (see 3.5.2.2, "The HTML for a Server-Initiated Session" on page 80).
   e. Select the **Edit JCL** pushbutton, enter the JCL shown in Figure 83 and save it.

```
//&USER&SUBCHAR  JOB (051846,XXXX,,,,N),'&USER',MSGCLASS=T,USER=&USER,
//   REGION=4096K,CLASS=A,NOTIFY=&USER
//GO  EXEC PROC=&PROCEDURE,ISPFUSR=&USER
//TSO.SYSTSIN DD *
  ISPSTART PANEL(ISR@PRIM) NEWAPPL(ISR) GUI(IP:&SERVIP:&SERVPORT) +
           GUIWEB(&APPLICATION,&TOKEN,0)
  END
```

Figure 83. JCL for Server-Initiated Session

A detailed description of the variables used in the JCL Figure 83 can be found in *OS/390 V2R5.0 ISPF Application Server Guide and Reference*, SC34-4619. We review some of them in 3.5.2.2, “The HTML for a Server-Initiated Session” on page 80 and 3.5.2.3, “Connecting the Applet to the Host Application” on page 82.

This JCL starts an ordinary ISPF session with the ISPF Primary Option Menu, but it is possible to start any ISPF application by choosing any other panel, CLIST or program as a parameter of the ISPSTART command.

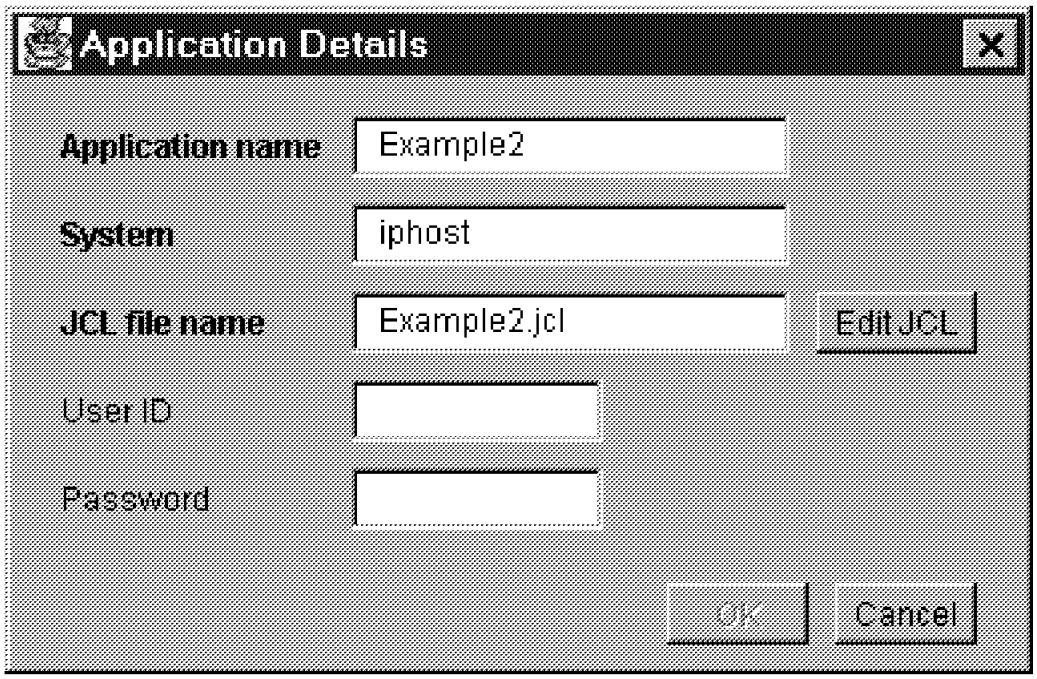


Figure 84. Application Details

3.5.2.2 The HTML for a Server-Initiated Session

For a Web browser that supports the required JRE level (see 3.3.1, “Software Prerequisites” on page 65), use the following HTML for the Workstation Agent Applet and save it as servbtch.htm in the HTML directory of the Web server.

If you wish to use Microsoft′s Internet Explorer, the HTML has to be changed according to the instructions in 3.5.3, “Using Sun′s Java Virtual Machine on Win32 Platforms” on page 83.

```
<HTML>
<TITLE> ISPF Workstation Agent Applet</TITLE>
<BODY>
<APPLET CODE="wsb.class"
ARCHIVE="jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar"
CODEBASE="http://ipserv/class" height="570" width="760">
<PARAM NAME=APPLICATION VALUE=Example2>
<PARAM NAME=AUTOCONNECT VALUE=n>
<PARAM NAME=BATCH VALUE=y>
<PARAM NAME=COMMAND VALUE=>
<PARAM NAME=MAXWAIT VALUE=30>
<PARAM NAME=PANELSINBROWSER VALUE=n>
<PARAM NAME=PASSWORD VALUE=>
<PARAM NAME=PORT VALUE=15994>
<PARAM NAME=PROCEDURE VALUE=TSOISPF>
<PARAM NAME=REQAPPLICATION VALUE=n>
<PARAM NAME=REQCOMMAND VALUE=n>
<PARAM NAME=REQPASSWORD VALUE=y>
<PARAM NAME=REQPROCEDURE VALUE=n>
<PARAM NAME=REQSYSTEM VALUE=n>
<PARAM NAME=REQUSER VALUE=y>
<PARAM NAME=RUNBUTTONTEXT VALUE=>
<PARAM NAME=SHOWOPTIONS VALUE=y>
<PARAM NAME=SYSTEM VALUE=>
<PARAM NAME=USER VALUE=>
</PARAM>
</APPLET>
</BODY>
</HTML>
```

Figure 85. HTML for Server-Initiated Session

The following is a review of some of the parameters of the Workstation Agent Applet:

- The value of CODEBASE is discussed in 3.3.3, “The ISPF Application Server Files” on page 68.
- The APPLICATION value must match the one you have chosen in Figure 84 on page 80.
- BATCH is now set to y, indicating that JCL is to be submitted to the host.
- The value of PORT equals the workstation port number, which was defined in Figure 77 on page 73.
- The PROCEDURE value must match the name of a suited cataloged procedure JCL on the host. For this example, use the one in Figure 80 on page 77.

  This value will be substituted for the &PROCEDURE variable in the JCL, which was defined in Figure 83 on page 79.
- The values of REQUSER and REQPASSWORD specify that after starting the application, the user is prompted for user ID and password.

All parameters are described in detail in *OS/390 V2R5.0 ISPF Application Server Guide and Reference*, SC34-4619.

3.5.2.3 Connecting the Applet to the Host Application

To build up the connection between the browser workstation and the application on the host, perform the following steps:

1. Enter the following URL in your Web browser (step **A:** in Figure 69 on page 63):

   `http://ipserv/servbtch.htm`

   The Workstation Agent Applet is loaded onto the browser workstation (step **B**) and starts running.

2. Enter your user ID and password when the applet prompts you for them.

3. Select the **Connect** pushbutton.

The connection to the Application Server will be established (step **C**). The Application Server transmits the JCL, which belongs to the application Example2, to the IP address iphost (step **D**) and JES starts the address space for the ISPF application (step **E**). The ISPSTART command in the JCL for the application (see Figure 83 on page 79) connects the host session to the Application Server (step **F**).

The values of the variables in this command (&SERVIP, &SERVPORT, &APPLICATION) will be substituted according to the definitions we have made this far. The &TOKEN variable is set to an unique value, which is generated by the Application Server to distinguish between different requests for this application.

The Application Server matches the two connections (step **G**). The connection is listed under Connections completed on the ISPF Application Server main window (Figure 86 on page 83), and the GUI interface starts on the browser workstation, in this example with the *ISPF Primary Option Menu, ISR@PRIM*.

Figure 86. ISPF Application Server-Connections Completed

3.5.3 Using Sun′s Java Virtual Machine on Win32 Platforms

As we have already stated, Microsoft′s Internet Explorer 4.0 does not support the required JRE level for executing the ISPF Workstation Agent Applet. A method to go around this problem is to use Sun′s Java Activator (now named "Java Plug-in"), which runs Java applets using Sun′s Java virtual machine (JVM) instead of the default one integrated in a Web browser. In this way you become independent of the Web browser, which is used to invoke the ISPF Workstation Agent Applet.

In this section we first describe the method of installing the Java Activator. Then we show how to change the HTML in Figure 78 on page 74 and Figure 85 on page 81 to invoke the Java Activator. In 3.5.3.2, "HTML for the Internet Explorer" on page 84, we present the HTML which only can be used with the Internet Explorer. If you wish to support both Internet Explorer and Netscape Navigator, use the HTML that is presented in 3.5.3.3, "HTML for Internet Explorer and Netscape Navigator" on page 85.

In any case, note the following:

Important

The HTML which we present in the following section is specific for use of the Java Activator at level Early Access Release 3. If you use a higher level (announced as Java Plug-In software), see the instructions on Suns′s Java Home Page for converting HTML.

3.5.3.1 Installing the Java Activator

The technology used to allow Sun′s JRE to run inside Internet Explorer is Micrososft′s COM/ActiveX. Using the HTML <OBJECT> tag, ActiveX controls or COM components can be run as part of a Web page.

To run Sun′s JRE inside Netscape Navigator, the plug-in architecture of the Navigator is used. The HTML <EMBED> tag allows plug-ins to be run as part of a Web page.

A more detailed description of the functionality of the Java Activator is available on Sun′s Java Web page.

To make use of the Java Activator, the applet HTML has to be changed to include the <OBJECT> or <EMBED> tags as previously mentioned. A free Java Activator HTML Converter, which executes these modifications, is available on Sun′s Java Internet pages. There is also a written specification to guide Web page authors to make these changes.

If a Web browser encounters for the first time a Web page that specifies the use of Java Activator, the download and installation process for the Java Activator is automatically started. The location from which the Activator files are downloaded is fixed in the HTML. Follow the instructions, which are displayed during the installation process, to complete the installation of the Java Activator on your browser workstation.

In subsequent encounters of Web pages that specify the use of Java Activator, it is invoked instantaneously from the hard drive of the workstation.

3.5.3.2 HTML for the Internet Explorer

We used the Java Activator HTML Converter at level Early Access Release 3 to convert the HTML for our first example (Figure 78 on page 74). The result (without some unnecessary comments) is shown in Figure 87 on page 85.

```
<HTML>
<TITLE> ISPF Workstation Agent Applet</TITLE>
<BODY>
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
WIDTH = "760" HEIGHT = "570"
codebase=
"http://java.sun.com/products/activator/ea3/jinstall-11-ea3-win32.cab#Version=1,0,3,0">
<PARAM NAME = CODE VALUE = "wsb.class" >
<PARAM NAME = CODEBASE VALUE = "http://ipserv/class" >
<PARAM NAME = ARCHIVE VALUE =
"jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar">

<PARAM NAME="type" VALUE="application/x-java-applet;version=1.1">
<PARAM NAME = APPLICATION VALUE =>
<PARAM NAME = AUTOCONNECT VALUE =n>
<PARAM NAME = BATCH VALUE =n>
<PARAM NAME = COMMAND VALUE =>
<PARAM NAME = MAXWAIT VALUE =30>
<PARAM NAME = PANELSINBROWSER VALUE =n>
<PARAM NAME = PASSWORD VALUE =>
<PARAM NAME = PORT VALUE =15994>
<PARAM NAME = PROCEDURE VALUE =>
<PARAM NAME = REQAPPLICATION VALUE =y>
<PARAM NAME = REQCOMMAND VALUE =n>
<PARAM NAME = REQPASSWORD VALUE =n>
<PARAM NAME = REQPROCEDURE VALUE =n>
<PARAM NAME = REQSYSTEM VALUE =n>
<PARAM NAME = REQUSER VALUE =n>
<PARAM NAME = RUNBUTTONTEXT VALUE =>
<PARAM NAME = SHOWOPTIONS VALUE =y>
<PARAM NAME = SYSTEM VALUE =>
<PARAM NAME = USER VALUE =>
</PARAM>
</OBJECT>
</BODY>
</HTML>
```

Figure 87. HTML for the Internet Explorer

The boldtype parameter defines the location from which the Java Activator files are downloaded.

The HTML for our second example, the server-initiated session (see Figure 85 on page 81), has to be converted in the same way. You can use the HTML in Figure 87 as a model and change the values of the following parameters in Figure 87:

```
<PARAM NAME = APPLICATION VALUE =Example2>
<PARAM NAME = BATCH VALUE =y>
<PARAM NAME = PROCEDURE VALUE =TSOISPF>
<PARAM NAME = REQPASSWORD VALUE =y>
<PARAM NAME = REQUSER VALUE =y>
```

3.5.3.3 HTML for Internet Explorer and Netscape Navigator

If the ISPF Workstation Agent Applet HTML should be prepared to be used by both Internet Explorer and Netscape Navigator, the <OBJECT> and the <EMBED> tag must be included in the new HTML.

Figure 88 on page 86 shows the HTML for this case, which we have generated with the help of the Java Activator HTML Converter from the HTML shown in Figure 78 on page 74.

In the first part (up to the <COMMENT> tag), we recognize the HTML tags we have already seen in 3.5.3.2, “HTML for the Internet Explorer” on page 84. The Internet Explorer ignores everything between the <COMMENT> and </COMMENT> tags.

It also ignores the </NOEMBED> and </EMBED> tags, since there are no corresponding <NOEMBED> and <EMBED> tags. So for the Internet Explorer, we obtain the same HTML as in Figure 87.

```
<HTML>
<TITLE> ISPF Workstation Agent Applet</TITLE>
<BODY>
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
WIDTH = "760" HEIGHT = "570"
codebase=
"http://java.sun.com/products/activator/ea3/jinstall-11-ea3-win32.cab#Version=1,0,3,0">
<PARAM NAME = CODE VALUE = "wsb.class" >
<PARAM NAME = CODEBASE VALUE = "http://ipserv/class" >
<PARAM NAME = ARCHIVE VALUE =
"jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar" >

<PARAM NAME="type" VALUE="application/x-java-applet;version=1.1">
<PARAM NAME = APPLICATION VALUE =>
<PARAM NAME = AUTOCONNECT VALUE =n>
<PARAM NAME = BATCH VALUE =n>
<PARAM NAME = COMMAND VALUE =>
<PARAM NAME = MAXWAIT VALUE =30>
<PARAM NAME = PANELSINBROWSER VALUE =n>
<PARAM NAME = PASSWORD VALUE =>
<PARAM NAME = PORT VALUE =15994>
<PARAM NAME = PROCEDURE VALUE =>
<PARAM NAME = REQAPPLICATION VALUE =y>
<PARAM NAME = REQCOMMAND VALUE =n>
<PARAM NAME = REQPASSWORD VALUE =n>
<PARAM NAME = REQPROCEDURE VALUE =n>
<PARAM NAME = REQSYSTEM VALUE =n>
<PARAM NAME = REQUSER VALUE =n>
<PARAM NAME = RUNBUTTONTEXT VALUE =>
<PARAM NAME = SHOWOPTIONS VALUE =y>
<PARAM NAME = SYSTEM VALUE =>
<PARAM NAME = USER VALUE =>
<COMMENT>
<EMBED type="application/x-java-applet;version=1.1"
java_CODE = "wsb.class"
java_CODEBASE = "http://ipserv/class"
java_ARCHIVE = "jar/ispfdt.jar,jar/wsb.jar,jar/ispfnls.jar,jar/ispfed.jar"
WIDTH = "760"
HEIGHT = "570"
AUTOCONNECT = n
BATCH = n
MAXWAIT = 30
PANELSINBROWSER = n
PORT = 15994
REQAPPLICATION = y
REQCOMMAND = n
REQPASSWORD = n
REQPROCEDURE = n
REQSYSTEM = n
REQUSER = n
SHOWOPTIONS = y
pluginspage="http://java.sun.com/products/activator/ea3/plugin-install.html">
<NOEMBED>
</COMMENT>
</PARAM>
</NOEMBED>
</EMBED>
</OBJECT>
</BODY>
</HTML>
```

Figure 88. HTML for Internet Explorer and Netscape Navigator

The Netscape Navigator, on the other hand, does not understand the <OBJECT> and <COMMENT> tags and ignores them: The Navigator does not read the tags

between the <BODY> and the first <EMBED> tag, it recognizes only those tags that allow it to make use of the Java Activator.

If you use the Java Activator HTML converter to generate this HTML, it is important that you reformat the part for the Navigator as shown in Figure 88 on page 86. It is not allowed to assign an empty value to a parameter, such as:

```
APPLICATION =
```

Delete all statements after the first <EMBED> tag, in which an empty value is assigned to a parameter.

To get the HTML of the server-initiated session (see Figure 85 on page 81) for using both Web browsers, change the following parameter in the Internet Explorer and the Navigator part of Figure 88 on page 86:

```
<PARAM NAME = APPLICATION VALUE =Example2>
<PARAM NAME = BATCH VALUE =y>
<PARAM NAME = PROCEDURE VALUE =TSOISPF>
<PARAM NAME = REQPASSWORD VALUE =y>
<PARAM NAME = REQUSER VALUE =y>
```

Chapter 4. VisualAge for ISPF

Prior to ISPF for OS/390 Version 2 Release 5, there were two ways of coding ISPF panels:

1. Through panel definition statements using the “traditional” panel language
2. Through the dialog Tag Language (DTL)

With ISPF for OS/390 Release 5, a third method of creating ISPF panels is introduced: VisualAge for ISPF.

VisualAge for ISPF provides a visual way of constructing panels from a desktop workstation. Most of the time, only a limited knowledge of the panel languages is required. VisualAge for ISPF generates as output a file with traditional panel definition statements, which can be transferred to the host. This generated output code runs as a 3270 panel or a workstation GUI panel.

VisualAge also allows you to modify existing ISPF panel source. To do so, the panel definition statements, *not* the DTL source, must be transferred to a workstation and imported into VisualAge for ISPF.

In this chapter, we show how to install and use this new feature of ISPF for OS/390 Version 2 Release 5.

4.1 Installing VisualAge for ISPF

Before using VisualAge for ISPF (VA for ISPF), it must be installed on a workstation with one of the following operating systems:

- Microsoft Windows 95
- Microsoft Windows NT
- OS/2 Version 3.0 or later

The workstation should be equipped with:

- 100MHz Pentium processor
- 16MB installed RAM (32MB recommended)
- 40MB hard disk space available for VisualAge for ISPF

4.1.1 Installation Steps on the Workstation

To install VisualAge for ISPF, the installation routine must be downloaded to your workstation. Choose one of the methods listed under ISPF option 3.7.2 (see Figure 89 on page 90).

```
                    VisualAge for ISPF Component install

Command ===>

Operating System - Network
   1. OS/2 - with ISPF C/S
   2. OS/2 - with File Transfer Protocol Server
   3. OS/2 - without File Transfer Protocol Server
   4. Windows 95 or Windows NT - with ISPF C/S
   5. Windows 95 or Windows NT - with File Transfer Protocol Server
   6. Windows 95 or Windows NT - without File Transfer Protocol Server
   7. Manual Install
```

Figure 89. Download the VisualAge for ISPF Installation Routine

On the host, the installation routine resides in the hlq.SISPVENU library, where hlq refers to the high level qualifiers of your ISPF data sets. There are two installation routines in this library, ISPVAOX for OS/2 and **ISPVAWX** for Windows 95/NT.

If you choose the **Manual Install** option in Figure 89, make sure that you use a binary download and that you have selected the correct member for your operating system.

The last step in the installation procedure is to invoke the installation routine:

1. Go to an MS-DOS command prompt.
2. Change to the VisualAge for ISPF target directory:

   ```
   c:\vaispf
   ```

3. Type `vainst`, then press Enter.

VAINST.EXE is a self-extracting executable program that generates the necessary files in the *current directory* on your workstation.

4.1.2 Preparing to Modify Existing ISPF Panels

As previously mentioned, you may use VisualAge for ISPF to modify existing panels. However, there is one major problem if you download existing panels to your workstation and import them into VisualAge for ISPF: The panels may contain (especially if DTL was used to code them) non-displayable hexadecimal codes to define attribute characters in the)ATTR section.

Usual file transfer utilities (for example, FTP) do not support a correct conversion of these characters from EBCDIC to ASCII and vice versa. To go around this problem, ISPF for OS/390 Release 5 includes a new file transfer utility, ISPF FILEXFER, which provides a one-to-one mapping of all 256 code points between the host and the workstation code page.

To make use of the FILEXFER service, perform the following steps:

1. An ISPF Workstation Agent (WSA) connection between the workstation and the host ISPF session has to established.

   To install, start and connect the ISPF WSA on your workstation, perform the following steps:

   a. Download the WSA installation procedure to your workstation as described in ISPF option 3.7.1. This procedure is very similar to the one

for VisualAge for ISPF in 4.1.1, “Installation Steps on the Workstation” on page 89.

b. Start the installation routine on your workstation.

c. Start the ISPF WSA on your workstation.

```
c:\wsa>wsa
```

d. Connect the host to the workstation by selecting 1. Workstation connection...from the Workstation action bar of the ISPF settings panel (Option 0) and complete the Initiate Workstation Connection panel, as shown in Figure 90.

```
                  Initiate Workstation Connection
Command ===>

Workstation Connection                    GUI Network Protocol
2  1. With GUI display                    1  1. TCP/IP
   2. Without GUI display                    2. APPC
   3. Connect to ISPF Application Server     3. Use ISPDTPRF file

GUI Title

TCP/IP Address
IP address of your workstation
APPC Address

Application Name


Maximum Connection Wait Time . . . 60

Host Codepage  . . . 37       Host Character Set . . . 697

GUI Window Frame              Default Window Background Color
1  1. Standard (STD)          1  1. Dialog (DLG)
```

Figure 90. Connecting a Host ISPF Session to the WSA

If APPC is used to connect to the workstation, you must change the panel in Figure 90 accordingly.

You may also use a connection with GUI display (Workstation Connection option 1). Note that a connection via the ISPF Application Server (option 3) does not support the FILEXFER service. (see Chapter 3, “Accessing ISPF from the World Wide Web” on page 63)

2. After connecting the WSA to the ISPF session, download the Transfer Map to the VisualAge for ISPF directory of your workstation. Use ISPF option 3.7.3. Complete the panel as shown in Figure 91.

```
                Download VisualAge for ISPF Transfer Map
Command ===>

  Directory to copy file to:
  c:\vaispf
  Data Set to copy file from:
  'hlq.SISPVENU'
```

Figure 91. Download of the ISPF Transfer Map

The Transfer Map needs to be downloaded only once.

3. Use ISPF option 3.7.4 to download existing ISPF panels to your workstation. Complete the panel like the one shown in Figure 92 where we download the ISPF primary panel:

```
                Download/Upload Data Set To/From Workstation
Command ===>

ISPF Library:
   Project . . .
   Group . . . .          . . .          . . .          . . .
   Type  . . . .
   Member  . . .               (Blank or pattern for member selection list)

Other Partitioned or Sequential Data Set:
   Data Set Name . . . 'hlq.SISPPENU(ISR@PRIM)'
   Volume Serial . . .         (If not cataloged)

Workstation File:
   File Name . . . . . c:\vaispf\samples\isrprim.isp

Download or upload                  Options
1  1. Download to workstation       /  Generate statistics on upload
   2. Upload from workstation       /  Transfer in text mode
```

Figure 92. Download Existing Panels to the Workstation

It is important to specify the **Transfer in text mode** option for existing panels in Figure 92.

Use the FILEXFER service in the same way to upload ISPF panels to the host.

4.2 Using VisualAge for ISPF

In this section, we show how to use VisualAge for ISPF as follows:

- Using VisuakAge for ISPF
- Using an example of creating a new ISPF panel with VisualAge for ISPF.

4.2.1 Using VisualAge for ISPF

To start VisualAge for ISPF, go to a DOS command prompt and change to the working directory of VisualAge for ISPF on your workstation which will be (c:\vaispf, if you followed the installation example in 4.1.1, “Installation Steps on the Workstation” on page 89.

Then type `vaispf` and press Enter.

The VisualAge for ISPF Quick Start Panel appears, as shown in Figure 93 on page 93.

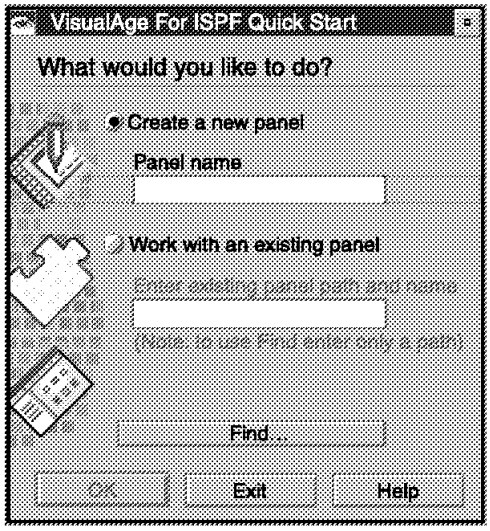


Figure 93. VisualAge for ISPF Composition Editor Panel

Select the option Create a new panel and type Test as panel name.

The next panel you receive is the Composition Editor window (see Figure 94).

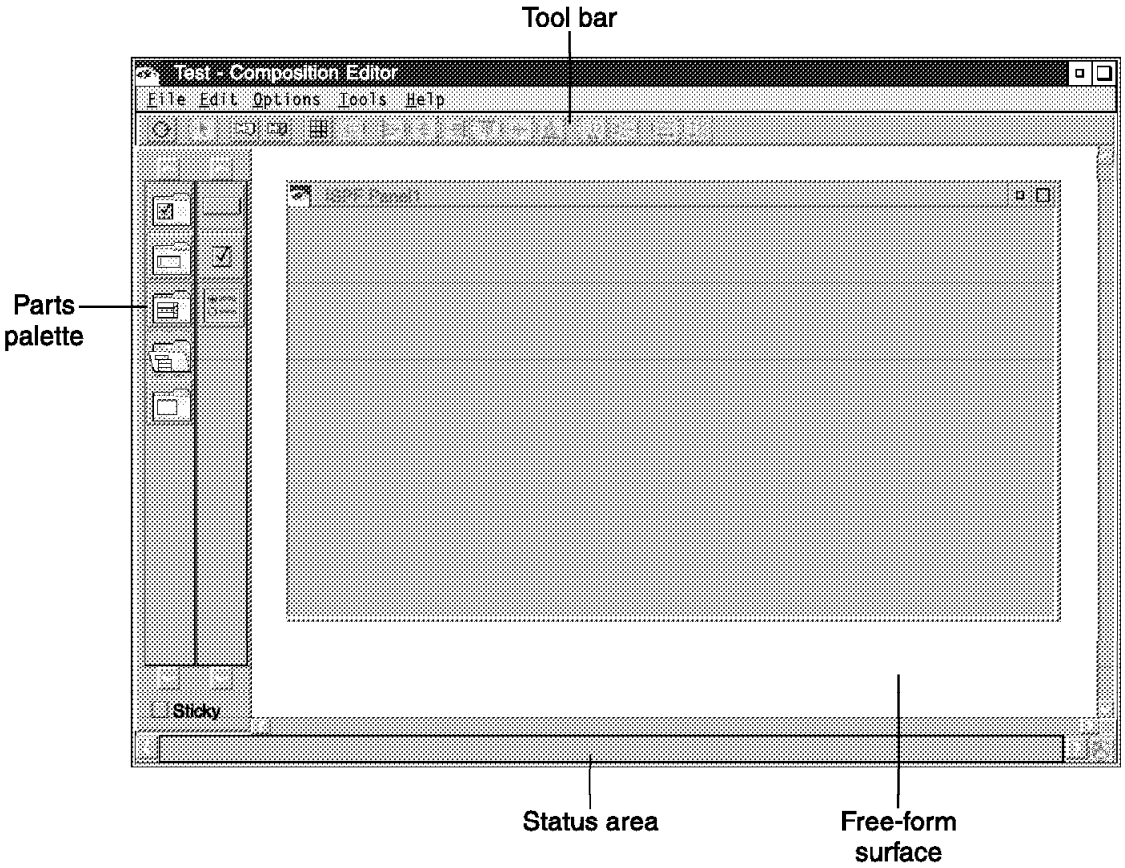


Figure 94. VisualAge for ISPF Composition Editor Panel

In Figure 94 the different areas of the Composition Editor panel are marked. On the free-form surface you can place the objects that are part of your ISPF panel.

The ISPF panel shell part, which can be seen here, is added by default to any new panel.

The parts you can put on the free-form surface may be selected from the parts palette, which is shown in detail in Figure 95.

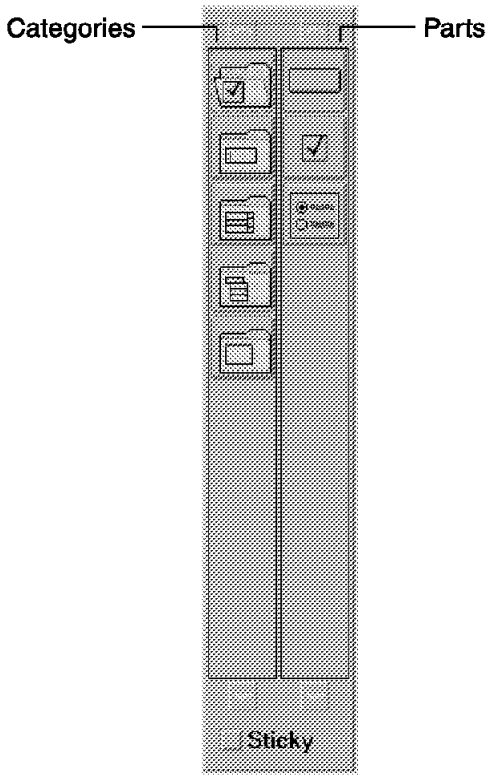


Figure 95. Parts Palette

The objects you can place on the free-form surface are classified in five *categories* In each category there are several available choices which can be used to lay out an ISPF panel. After you select one of the categories in Figure 95, the parts palette changes to show the available parts for this category.

Table 1 summarizes all categories and their parts:

Table 1 (Page 1 of 2). Categories and Parts

Category	Part	Description
Buttons	Push Button	Provides a choice that is activated when a user selects it.
	Toggle Button	Provides a settings choice with two clearly distinguishable selections. This is also known as a *check box*.
	Radio Button Set	Provides a set of mutually exclusive choices from which a user can select only one.
Data Entry	Text	Provides a field that enables a user to type a single line of text in a specified format.
	Label	Displays static text or graphic information.

Table 1 (Page 2 of 2). Categories and Parts		
Category	**Part**	**Description**
Lists	List	Provides a list of items from which a user can select only a single item.
	Drop-down list	Provides a hidden list of items from which a user can select only a single item.
	Combo Box	Provides a text field and a hidden list. The text field enables a user to type a value, and the list enables a user to select an item.
Menus	MenuBar	Provides a menu bar to contain menu bar items, and puts the first menu bar item in place on the menu bar.
	MenuBar Item	Provides a menu bar item that displays a pull-down menu when the user selects it.
	Menu Choice	Provides a menu choice for a menu bar item.
	Separator	Displays a graphical line for separating menu choices.
Canvas	ISPF Panel	Provides a panel with a frame border that can contain other visual parts.
	Group Box	Provides an area that displays a rectangular box for grouping related visual parts.
	Scrolled Window	Provides an area that can be scrolled by the user and can contain other visual parts.

The check box labeled Sticky in the lower left of the parts palette enables you to add several parts of the same type from the parts palette to the free-form surface without reselecting the part again after every placement.

The layout of every part on the free-form surface is defined through a set of parameters called *attributes*. These attributes can be changed in the General tab of the settings pages of a part. To display the settings pages on the screen, place the mouse pointer on the part. Then there are two ways of proceeding:

- Double-click mouse button 1.

or

- Click mouse button 2, and then select **Open settings** from the pop-up menu that appears.

A summary of the attributes of each part can be found in *OS/390 V2R5.0 VisualAge for ISPF User*, SC34-4620

Table 2 lists and describes the Tools included in VisualAge for ISPF, which can be selected either from the Tools option of the menu bar or from the icons on the Tool bar.

Table 2 (Page 1 of 2). Tools	
Tool	**Description**
Test	Show an approximation of what the current panel will look like when finished. Parts such as drop-down lists should work in the test panel as well.
Selection Tool	Release the part that has been selected from the palette, so **no** is selected.

Table 2 (Page 2 of 2). Tools	
Tool	**Description**
Show Connections	Show the connection lines between parts that are connected.
Hide Connections	Do not show the connecting lines between parts that are connected.
Toggle Grid	Toggle on or off the grid that underlies the free-form area (some parts also support grids). This can be helpful in spacing parts.
Snap to Grid	Place the edge of a part on the closest grid line to it when placed.
Align Left	Align the marked parts at the left-most point of the anchor part.
Align Center	Align the marked parts using the center of the anchor part as the alignment point.
Align Right	Align the marked parts at the right-most point of the anchor part.
Align Top	Align the marked parts at the top-most point of the anchor part.
Align Middle	Align the marked parts using the center of the anchor part as the alignment point.
Align Bottom	Align the marked parts at the bottom-most point of the anchor part.
Distribute Horizontally	Spread the marked parts evenly within the frame in the free-form area.
Distribute Vertically	Spread the marked parts evenly within the frame in the free-form area.
Match Width	Make all of the marked parts as wide as the anchor part.
Match Height	Make all of the marked parts as tall as the anchor part.

4.2.2 An Example of Creating an ISPF Panel

In order to learn how to work with VisualAge for ISPF, in this section we present a detailed example for creating a new ISPF panel. The example is to redesign the CUASELC panel of the Hotel Selector Dialog Application described in *OS/390 V2R5.0 ISPF Examples*, SC28-1282. The panel in 3270 mode is shown in Figure 96 on page 97.

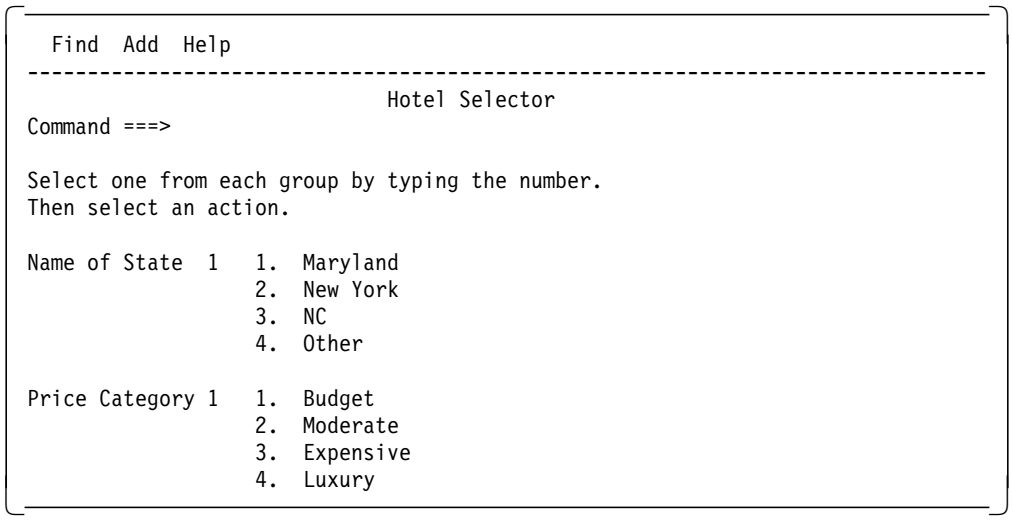


Figure 96. The CUASELC Panel of the Hotel Selector Dialog Application

1. Start VisualAge for ISPF.
   a. Proceed as described in 4.2.1, “Using VisualAge for ISPF” on page 92.
   b. Choose the option **Create a new panel** and select **Test** as panel name (see Figure 93 on page 93).
2. Add the MenuBar.
   a. Click mouse button 1 on the **Menus** category (the 4th on the category bar).
   b. Click on the **MenuBar** part (the 1st on the parts bar). The icon “greys” to show that it is selected.
   c. Move the cursor to the top of the panel in the free-form surface and click mouse button 1 again.

   Now the composition editor surface looks like the one shown in Figure 97 on page 98.

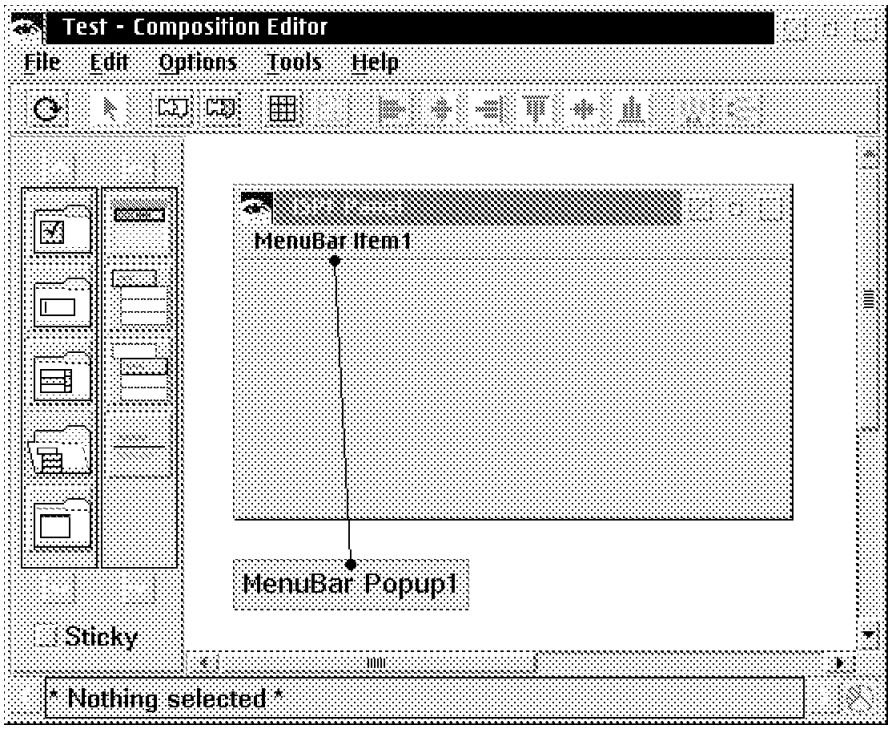


Figure 97. Add the MenuBar

3. Change the MenuBar Item1 settings.
   a. Place the cursor on MenuBar Item1 and double-click mouse button 1.
   b. Complete the window in Figure 98 like shown.

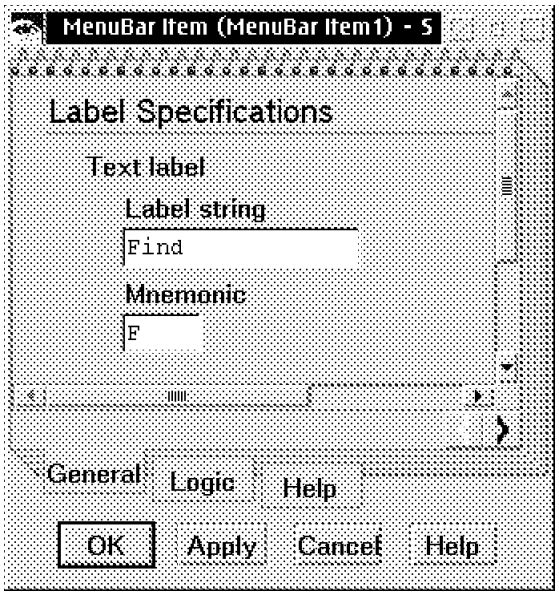


Figure 98. Change the MenuBar Item1 Attributes

   c. Click on the **Logic** tab and complete the Initialization Logic and the Processing Logic panels as indicated in Figure 99 on page 99 and Figure 100 on page 99.
   d. Click on **OK**.

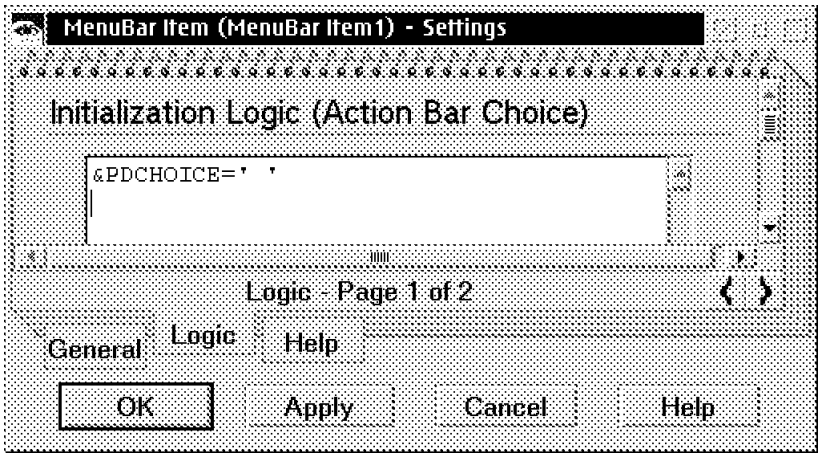


Figure 99. Initialization Logic Window

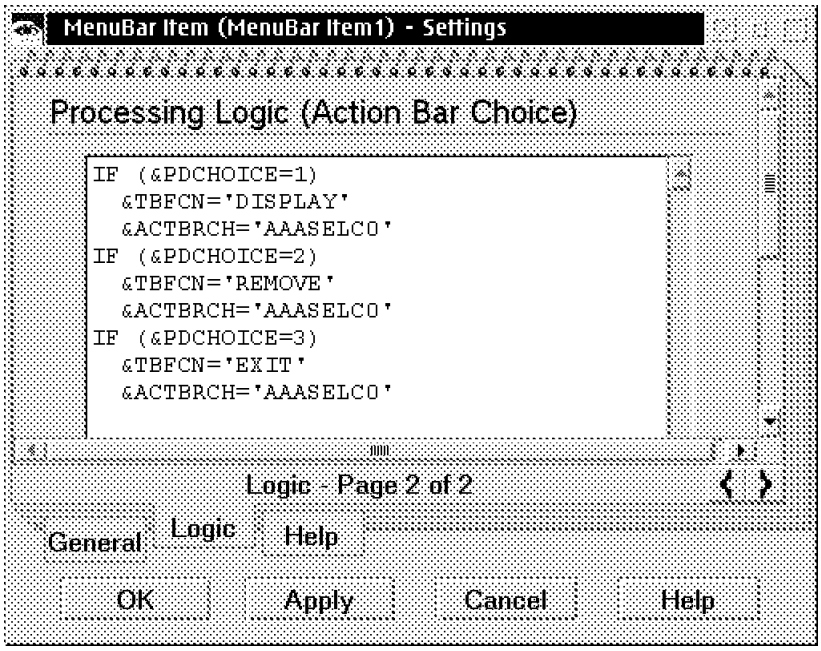


Figure 100. Processing Logic Window

4. Add the first Menu Choice to the MenuBar Popup1.
   a. Click on the **Menu Choice** part (the 3rd on the parts bar).
   b. Move the cursor to MenuBar Popup1 and click on mouse button 1.
5. Change the settings of Menu Choice1.
   a. Place the cursor on Menu Choice1 and double-click mouse button 1.
   b. Complete the windows shown in Figure 101 on page 100 and Figure 102 on page 100 as indicated. Quit the window by clicking on **OK**.

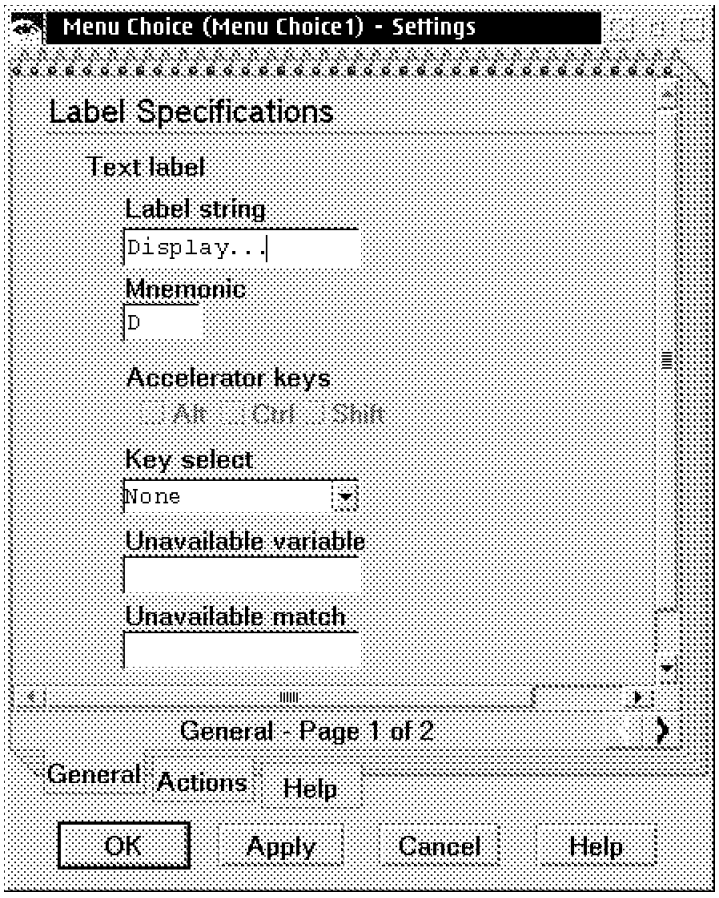


Figure 101. The MenuChoice1 Attributes

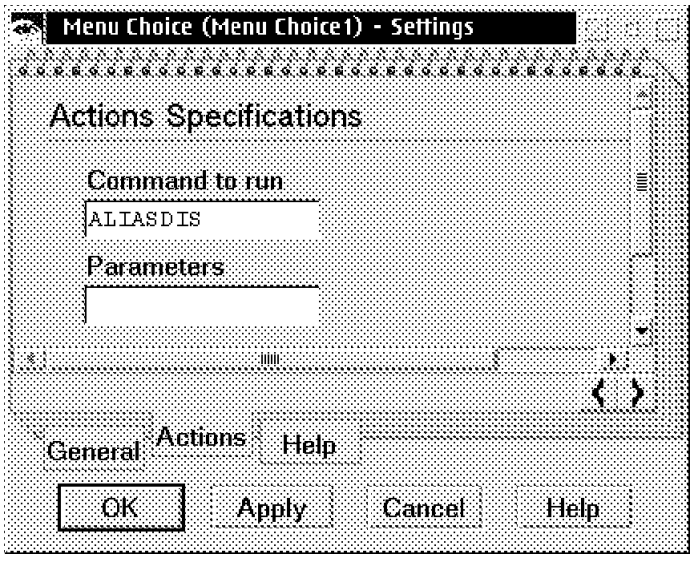


Figure 102. The MenuChoice1 Actions Specifications

6. Add the second and third Menu Choice to the MenuBar Popup1 and change their settings.

   Proceed in the same manner as in step 4 on page 99 and step 5 on page 99. Enter the settings listed in Table 3 on page 101.

Table 3. Settings for the 2nd and 3rd Choice of MenuBar Popup1			
Choice	**Label String**	**Mnemonic**	**Command to run**
Menu Choice2	Remove...	R	PASSFIND
Menu Choice3	Exit	E	PASSEXIT

7. Add the second MenuBar Item to the MenuBar and change its settings.
   a. Click on the **MenuBar Item** (the 2nd in the parts palette).
   b. Place the cursor on the MenuBar in the free-form surface and click on mouse button 1.
   c. Double-click on MenuBar Item2 and change the Label string to *Add* and the Mnemonic to *A*.
   d. Click on the **Logic** tab and Add the following to the Initialization Logic window:

      ```
      &PDCHOICE=' '
      ```

   e. Add the following to the Processing Logic window:

      ```
      IF (&PDCHOICE=1)
        &TBFCN='ADDONE'
        &ACTBRCH='AAASELC1'
      IF (&PDCHOICE=2)
        &TBFCN='ADDMANY'
        &ACTBRCH='AAASELC1'
        &PLLDNCH='2'
      IF (&PDCHOICE=3)
        &TBFCN='ADDMOTEL'
        &ACTBRCH='AAASELC1'
        &PLLDNCH='3'
      ```

8. Add the following choices to MenuBar Popup2, as displayed in step 4 on page 99 and step 5 on page 99:

Table 4. Choices for MenuBar Popup2			
Choice	**Label String**	**Mnemonic**	**Command to run**
Menu Choice4	One hotel...	O	SETADD
Menu Choice5	Many in the same city...	M	
Menu Choice6	Many with the same name...	a	

9. Add the third MenuBar Item like the second one in step 7.
   a. The Label String is *Help* and the Mnemonic is *H*.
   b. Make the following entry for the Initialization Logic:

      ```
      &PDCHOICE=' '
      ```

   c. Enter this as Processing Logic:

      ```
      IF (&PDCHOICE=1) &ACTBRCH='AAASELC3'
      ```

10. Add the following choices to MenuBar Popup3 as shown in step 4 on page 99 and step 5 on page 99:

Table 5. Choices for MenuBar Popup3			
Choice	**Label String**	**Mnemonic**	**Command to run**
Menu Choice7	How to get help...	H	
Menu Choice8	Extended help...	E	EXHELP
Menu Choice9	Keys help...	K	KEYSHELP

If you hide the connections, now the composition editor looks like the one shown in Figure 103.

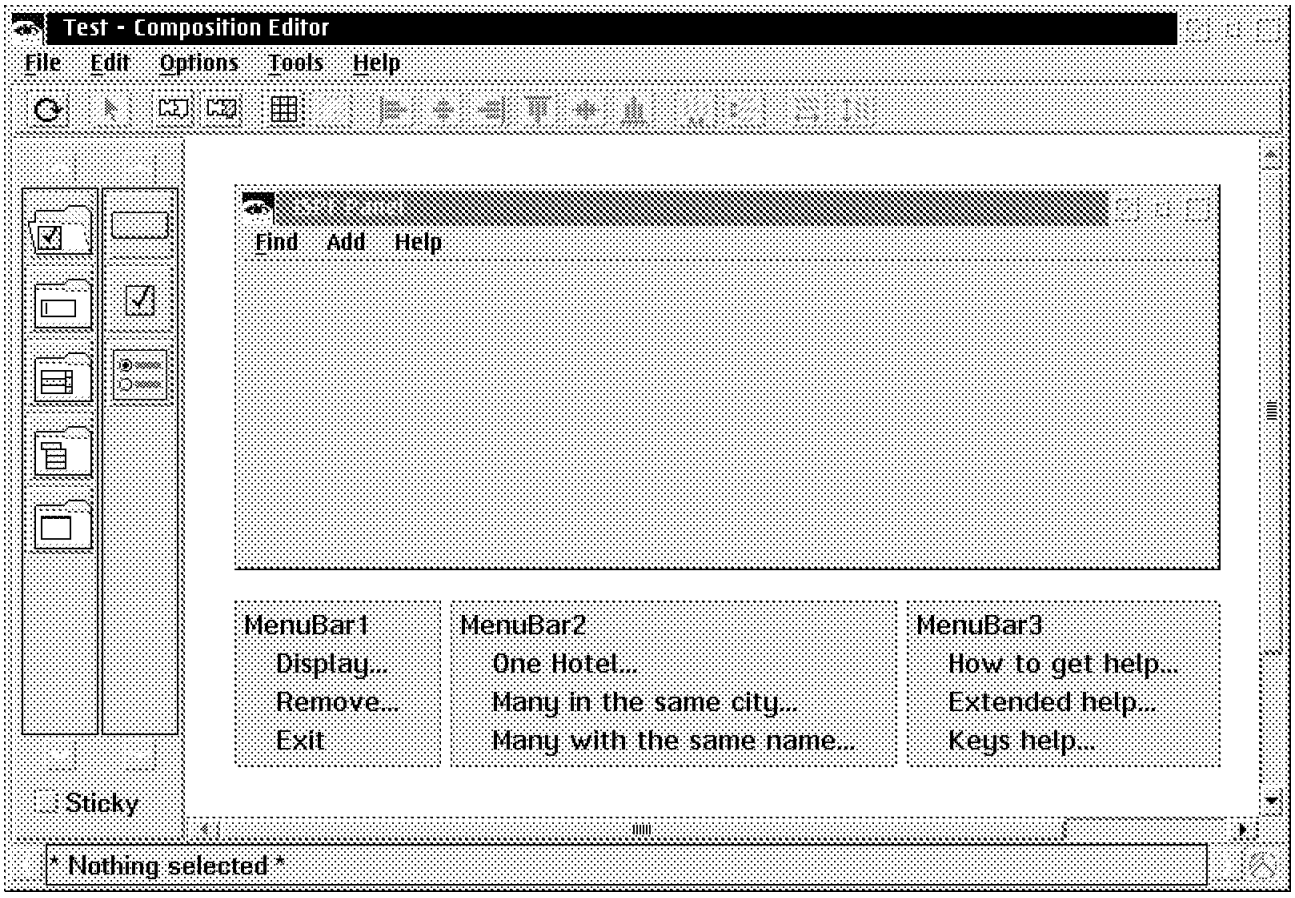


Figure 103. Composition Editor after Finishing the MenuBar

11. Add the panel title.
    a. Click on the **Data Entry** category (the 2nd of the category bar).
    b. Click on the **Label** part (the 2nd of the parts bar).
    c. Move the cursor to the position of the panel title on the free-form surface and click on mouse button 1.
    d. Change the attributes of *Label1* as shown in Figure 104 on page 103.

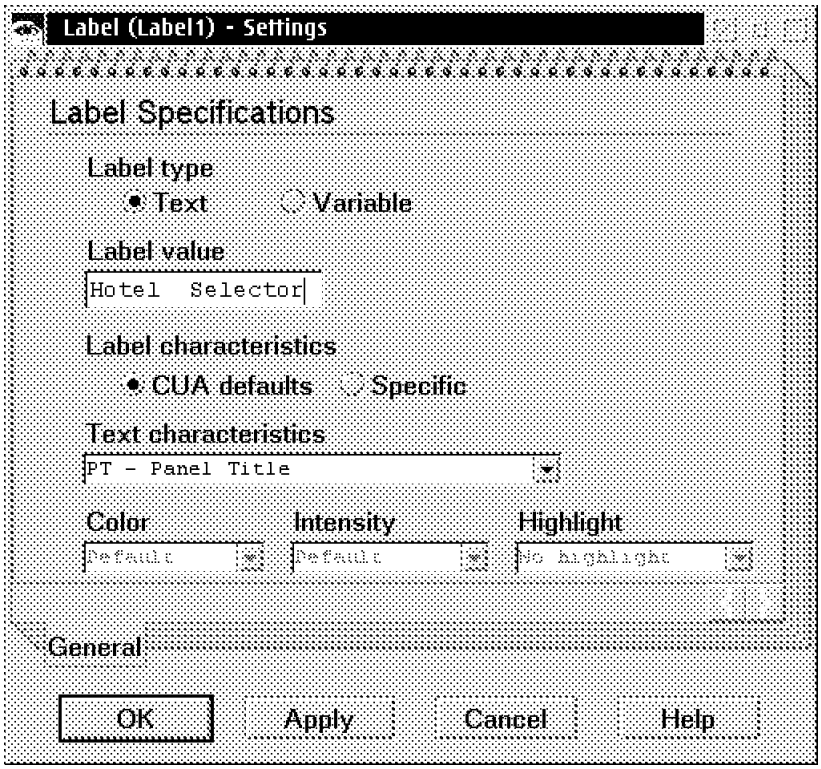


Figure 104. Attributes of the Panel Title

12. In the same manner, add the following text strings:

Table 6. Text Strings

Label	Label value	Text characteristics
Label2	Command = = = >	FP - Field Prompt
Label3	Select one from each group by typing the number.	NT - Normal Text
Label4	Then select an action.	NT - Normal Text
Label5	Name of State	FP - Field Prompt
Label6	Price Category	FP - Field Prompt

If necessary, use the tools to align the parts on the panel.

The composition editor now should look like in Figure 105 on page 104.

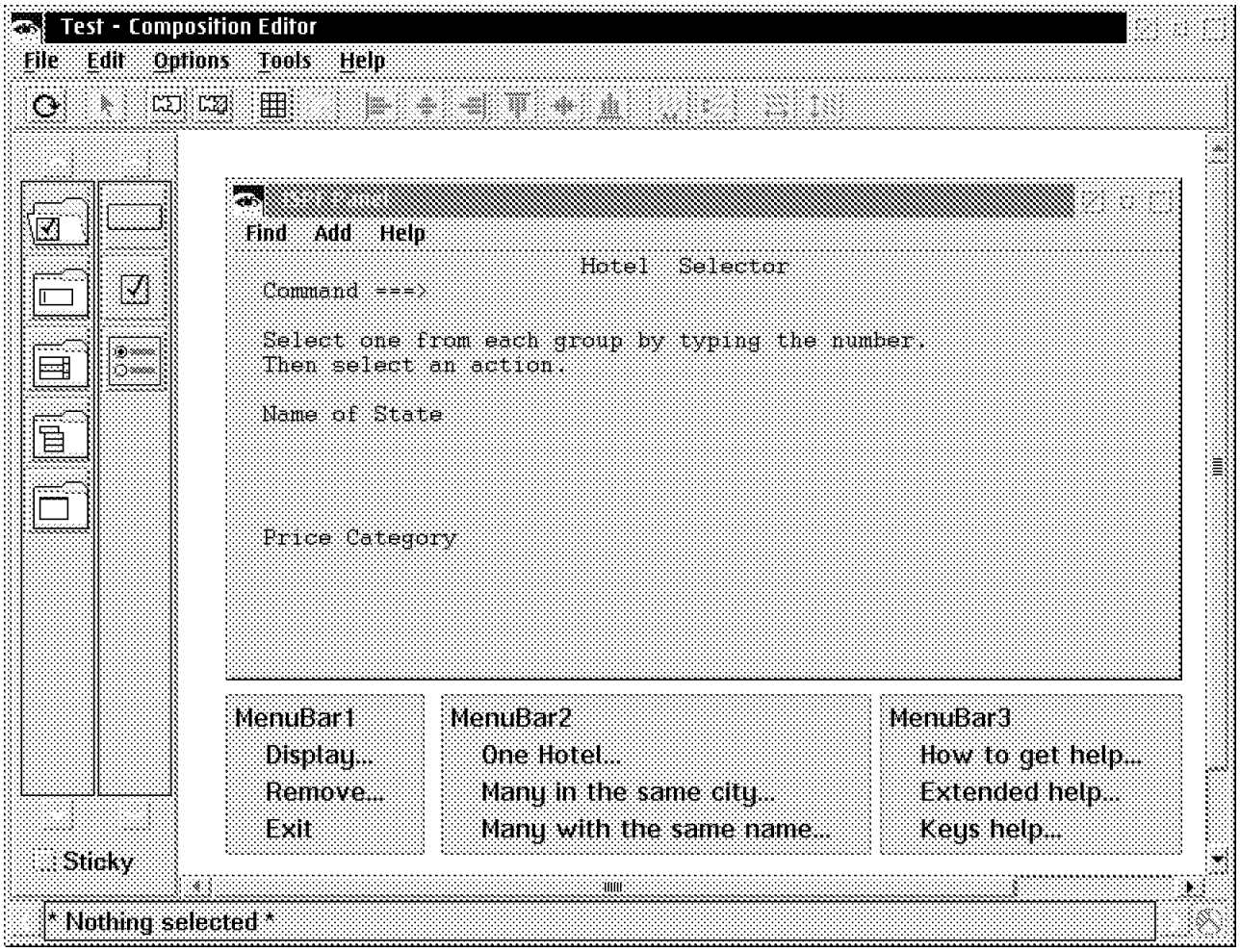


Figure 105. Composition Editor after Finishing Text Labels

13. Add the command line.

    a. Click on the **Text** part (the 1st one on the parts palette).

    b. Move the cursor to the right of the Command ===> label and click on mouse button 1.

    c. Change the attributes of the command line as shown in Figure 106 on page 105.

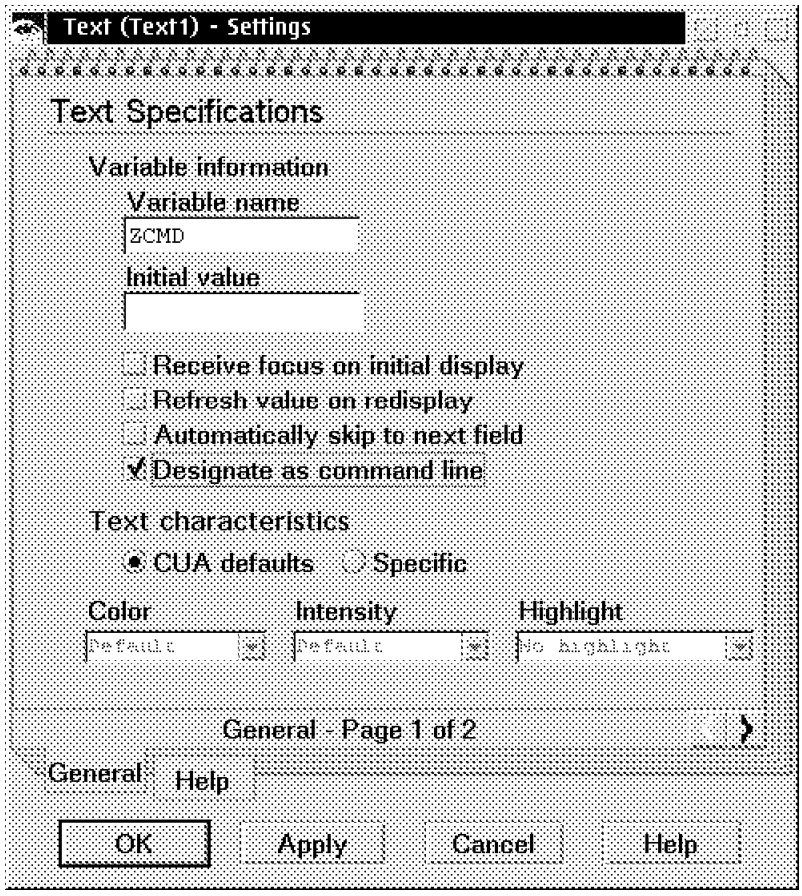


Figure 106. Attributes of the Command Line

14. Add the Name of State list:.
    a. Click on the **Lists** category (the 3rd on the category bar).
    b. Click on the **List** part (the 1st on the parts palette).
    c. Move the cursor to the right of the Name of State text and click on mouse button 1.
15. Change the settings of List1.
    a. Place the cursor on the added list and double-click mouse button 1.
    b. Complete the window as shown in Figure 107 on page 106.

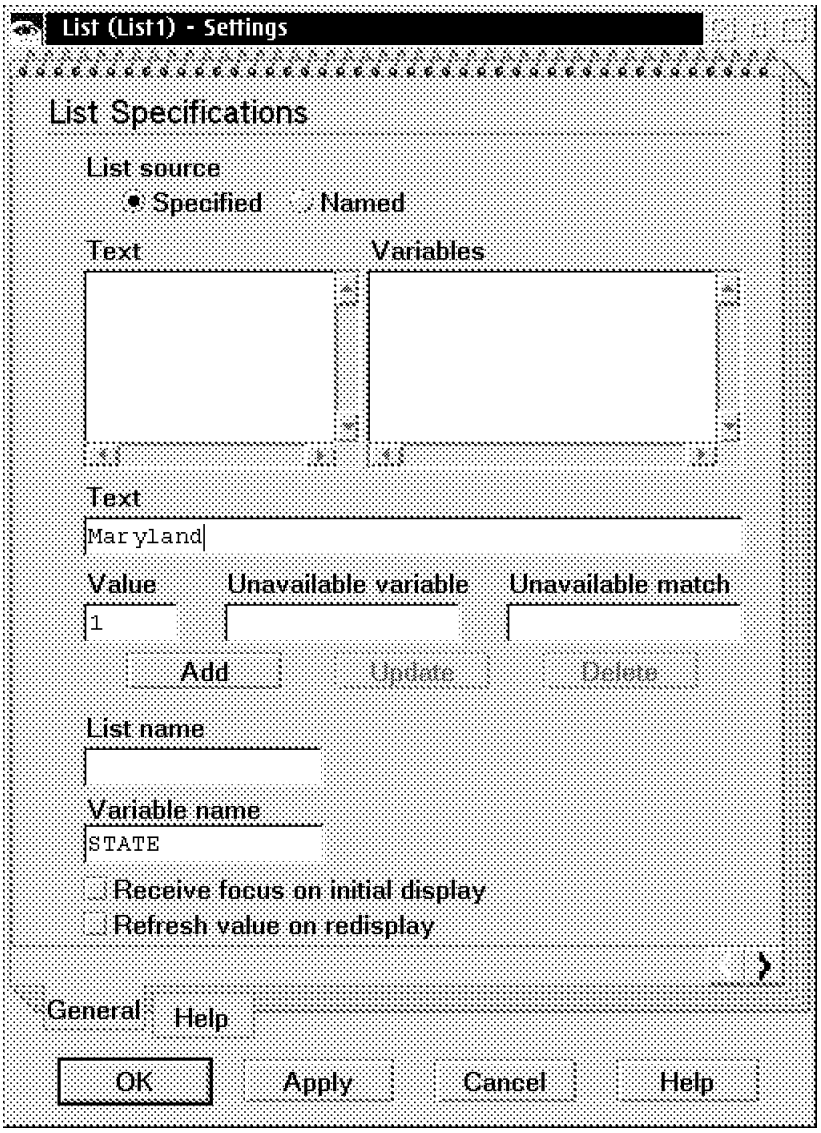


Figure 107. Add a List Item

c. Click on **Add**.

d. Enter in the same manner the other list items (New York, NC, Other). The List1 Settings now looks like the one shown in Figure 108 on page 107. Click on **OK**.

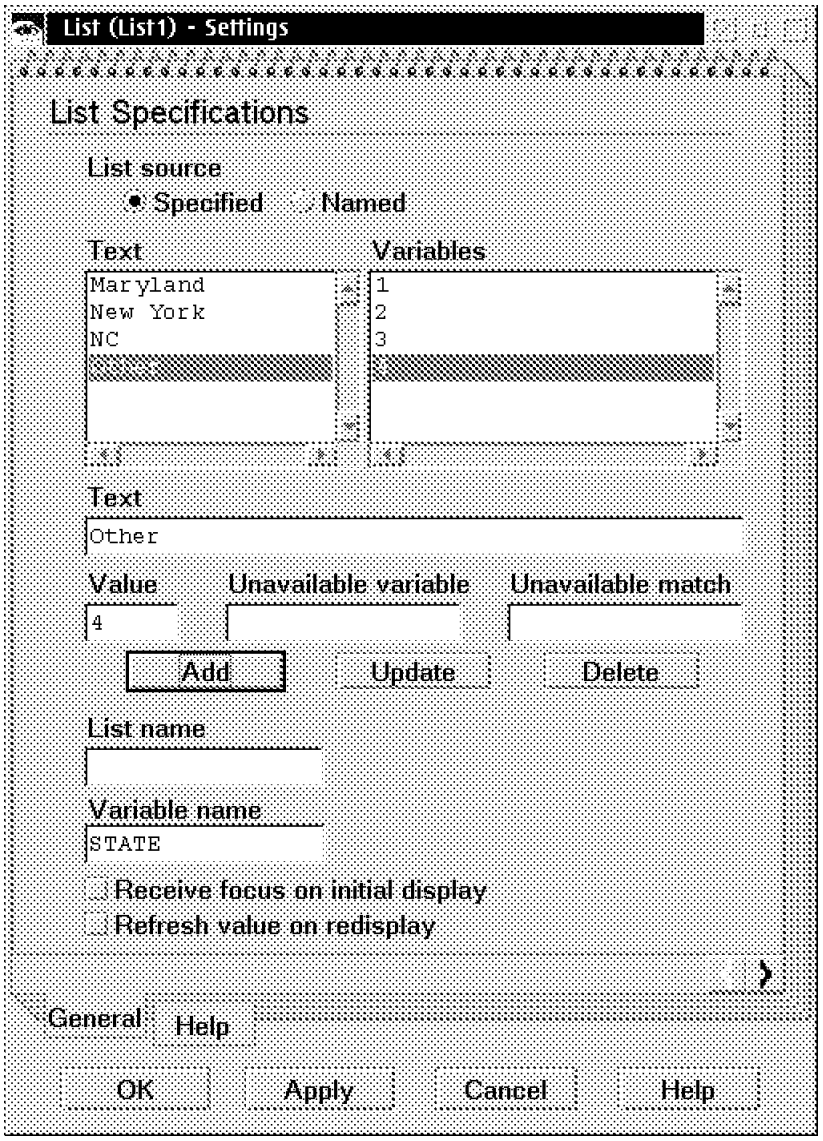


Figure 108. List1 Settings

16. Add the Price Category list in the same way as described in step 14 on page 105 and step 15 on page 105.
    a. The price categories are Budget, Moderate, Expensive, Luxury.
    b. The Variable name is AMOUNT.
17. Complete the settings of the ISPF Panel.
    a. Double-click anywhere on the panel, where no other parts are.
    b. Complete the General-Page 2 of 3 window as shown in Figure 109 on page 108.

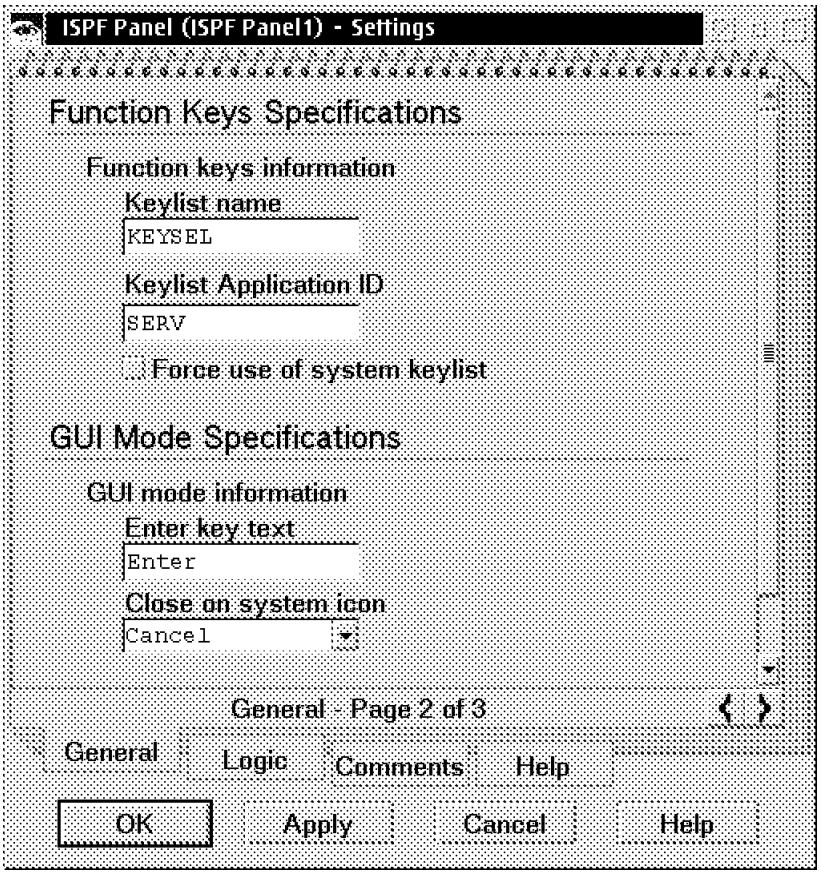


Figure 109. General Settings of the ISPF Panel - Page 2 of 3

c. Click on the **Logic** tab and enter the following text on the Initialization Logic panel:

```
 .ZVARS = '(ZCMD STATE AMOUNT)'
 .HELP = CUAXHLP
&ZCMD = ' '
&STATE = ' '
&AMOUNT = ' '
IF (&XSTE='MARYLAND') &STATE='1'
 .ATTR(AMOUNT)='CSRGRP(98) LISTBOX(ON) WIDTH(41) DEPTH(3)'
IF (&XSTE='NEW YORK') &STATE='2'
IF (&XSTE='NC') &STATE='3'
IF (&XSTE='OTHER') &STATE='4'
IF (&XAMT='BUDGET') &AMOUNT='1'
IF (&XAMT='MODERATE') &AMOUNT='2'
IF (&XAMT='EXPENSIVE') &AMOUNT='3'
IF (&XAMT='LUXURY') &AMOUNT='4'
```

d. Enter the following on the Processing Logic panel:

```
&STATE = TRANS(&STATE 01,1 02,2 03,3 04,4 *,*)
VER(&STATE RANGE,1,4)
IF (&STATE='1') &XSTE='MARYLAND'
IF (&STATE='2') &XSTE='NEW YORK'
IF (&STATE='3') &XSTE='NC'
IF (&STATE='4') &XSTE='OTHER'
&AMOUNT = TRANS(&AMOUNT 01,1 02,2 03,3 04,4 *,*)
VER(&AMOUNT RANGE,1,4)
IF (&AMOUNT='1') &XAMT='BUDGET'
```

```
IF (&AMOUNT='2') &XAMT='MODERATE'
IF (&AMOUNT='3') &XAMT='EXPENSIVE'
IF (&AMOUNT='4') &XAMT='LUXURY'
```

18. If necessary, resize and move the parts on your panel so that the composition editor now looks similar to the one in Figure 110.

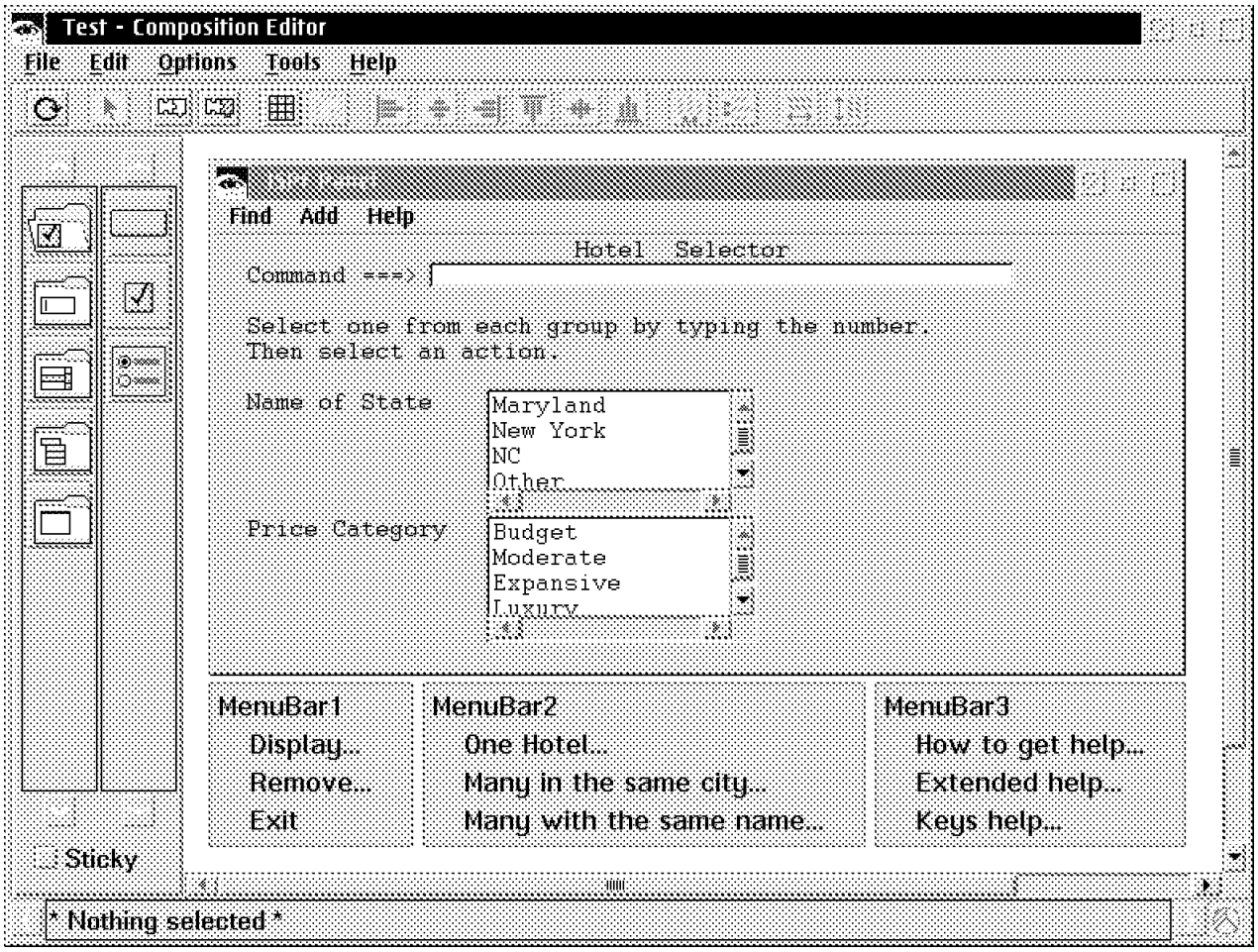


Figure 110. Final Look of the Composition Editor

19. Save the panel to the workstation hard drive.

    Use the **Save as...** option of the File menu bar item. As file name you may select c:\vaispf\samples\test.isp.

    VA for ISPF now generates the following runtime code:

```
)PANEL KEYLIST(KEYSEL,SERV)
)ATTR
 } TYPE(AB)
 { TYPE(ABSL) GE(ON)
 $ TYPE(PT)
 ! TYPE(FP)
 ? TYPE(NEF)
 # TYPE(CEF) LISTBOX(LB1) CSRGRP(1) WIDTH(19) DEPTH(5)
   TYPE(SAC) LISTBOX(ON) CSRGRP(1)
 @ TYPE(SAC)
 ] TYPE(CEF) LISTBOX(LB2) CSRGRP(2) WIDTH(19) DEPTH(5)
 * TYPE(SAC) LISTBOX(ON) CSRGRP(2)
)ABC DESC('Find') MNEM(1)
 PDC DESC('Display...')
 ACTION RUN(ALIASDIS)
 PDC DESC('Remove...')
```

```
 ACTION RUN(PASSFIND)
 PDC DESC('Exit')
 ACTION RUN(PASSEXIT)
)ABCINIT
 .ZVARS=PDCHOICE
&PDCHOICE=' '
)ABCPROC
IF (&PDCHOICE=1)
  &TBFCN='DISPLAY'
  &ACTBRCH='AAASELCO'
IF (&PDCHOICE=2)
  &TBFCN='REMOVE'
  &ACTBRCH='AAASELCO'
IF (&PDCHOICE=3)
  &TBFCN='EXIT'
  &ACTBRCH='AAASELCO'
)ABC DESC('Add') MNEM(1)
 PDC DESC('One Hotel...')
 ACTION RUN(SETADD)
 PDC DESC('Many in the same city...')
 PDC DESC('Many with the same name...')
)ABCINIT
 .ZVARS=PDCHOICE
&PDCHOICE=' '
)ABCPROC
IF (&PDCHOICE=1)
  &TBFCN='ADDONE'
  &ACTBRCH='AAASELC1'
IF (&PDCHOICE=2)
  &TBFCN='ADDMANY'
  &ACTBRCH='AAASELC1'
  &PLLDNCH='2'
IF (&PDCHOICE=3)
  &TBFCN='ADDMOTEL'
  &ACTBRCH='AAASELC1'
  &PLLDNCH='3'
)ABC DESC('Help') MNEM(1)
 PDC DESC('How to get help...')
 PDC DESC('Extended help...')
 ACTION RUN(EXHELP)
 PDC DESC('Keys help...')
 ACTION RUN(KEYSHELP)
)ABCINIT
 .ZVARS=PDCHOICE
&PDCHOICE=' '
)ABCPROC
IF (&PDCHOICE=1) &ACTBRCH='AAASELC3'
)BODY WINDOW(68,19) CMD(ZCMD)
+} Find} Add} Help+
{-------------------------------------------------------------------
                        $Hotel  Selector+
 !Command ===>?Z                                          +

 +Select one from each group by typing the number.+
 +Then select an action.+

 !Name of State+  #Z 1.@Maryland+
                     2.@New York+
                     3.@NC      +
                     4.@Other   +

 !Price Category+ ]Z*1.@Budget   +
                   *2.@Moderate +
                   *3.@Expansive+
                   *4.@Luxury   +


)INIT /* VAISPF GENLOGIC(3) */
 &ZWINTTL = 'ISPF Panel'
 .ZVARS = '(ZCMD STATE AMOUNT)'
 &ZCMD = ' '
 .ZVARS = '(ZCMD STATE AMOUNT)'
 .HELP = CUAXHLP
&ZCMD = ' '
&STATE = ' '
```

```
&AMOUNT = ' '
IF (&XSTE='MARYLAND') &STATE='1'
 .ATTR(AMOUNT)='CSRGRP(98) LISTBOX(ON) WIDTH(41) DEPTH(3)'
IF (&XSTE='NEW YORK') &STATE='2'
IF (&XSTE='NC') &STATE='3'
IF (&XSTE='OTHER') &STATE='4'
IF (&XAMT='BUDGET') &AMOUNT='1'
IF (&XAMT='MODERATE') &AMOUNT='2'
IF (&XAMT='EXPENSIVE') &AMOUNT='3'
IF (&XAMT='LUXURY') &AMOUNT='4'
)REINIT /* VAISPF GENLOGIC(0) */

)PROC
&STATE = TRANS(&STATE 01,1 02,2 03,3 04,4 *,*)
VER(&STATE RANGE,1,4)
IF (&STATE='1') &XSTE='MARYLAND'
IF (&STATE='2') &XSTE='NEW YORK'
IF (&STATE='3') &XSTE='NC'
IF (&STATE='4') &XSTE='OTHER'
&AMOUNT = TRANS(&AMOUNT 01,1 02,2 03,3 04,4 *,*)
VER(&AMOUNT RANGE,1,4)
IF (&AMOUNT='1') &XAMT='BUDGET'
IF (&AMOUNT='2') &XAMT='MODERATE'
IF (&AMOUNT='3') &XAMT='EXPENSIVE'
IF (&AMOUNT='4') &XAMT='LUXURY'
)LIST LB1
 VAL(1) CHOICE('Maryland')
 VAL(2) CHOICE('New York')
 VAL(3) CHOICE('NC')
 VAL(4) CHOICE('Other')
)LIST LB2
 VAL(1) CHOICE('Budget')
 VAL(2) CHOICE('Moderate')
 VAL(3) CHOICE('Expansive')
 VAL(4) CHOICE('Luxury')
)END /* VAISPF GENLEVEL(OS/390 R5 - BASE) MODLEVEL(OS/390 R5 - BASE) */
```

20. Upload the code to the host.

    Use the ISPF FILEXFER service (option 3.7.4) as displayed in Figure 111.

```
                Download/Upload Data Set To/From Workstation
Command ===>

ISPF Library:
   Project . . .
   Group . . . .          . . .          . . .          . . .
   Type  . . . .
   Member  . . .               (Blank or pattern for member selection list)

Other Partitioned or Sequential Data Set:
   Data Set Name . . . your.panel.library(test)
   Volume Serial . . .         (If not cataloged)

Workstation File:
   File Name . . . . . c:\vaispf\samples\test.isp

Download or upload                    Options
2  1. Download to workstation         /  Generate statistics on upload
   2. Upload from workstation         /  Transfer in text mode
```

Figure 111. Upload of the Generated Code

Now we have redesigned the CUASELC panel of the Hotel Selector Dialog Application in *OS/390 V2R5.0 ISPF Examples*, SC28-1282. If you want to test it, integrate it in the application by replacing the original CUASELC panel.

Chapter 5. Introduction to Firewall Technologies

Firewall technologies have been in use on other operating system platforms for some years, but to the OS/390 community, firewall technology is assumed to be a new area that is not commonly known or understood in detail.

This chapter explains what firewall technologies are, how each individual technology component works, and when to use which component.

A basic understanding of the TCP/IP protocol suite is assumed. Refer to Appendix C, "A Short Introduction to TCP/IP and the Internet" on page 199 for an introduction to TCP/IP. If you need more in-depth information on the TCP/IP protocol suite, you may find useful information in *TCP/IP Tutorial and Technical Overview*, GG24-3376.

This chapter includes the following topics:

- 5.1, What is a Firewall?
- 5.2, Firewall Categories
- 5.3, The OS/390 Firewall Technology Kit
- 5.4, IP Filtering
- 5.5, Network Address Translation (NAT)
- 5.6, Virtual Private Network (VPN)
- 5.7, Proxy Applications
- 5.8, The Socks Server
- 5.9, Domain Name Resolution (DNS)

For a detailed description of firewall technology on OS/390, see *Stay Cool on OS/390: Installing Firewall Technology*, SG24-2046.

5.1 What is a Firewall?

A *firewall* machine is a computer you use to separate a safe network from a not-so-safe network as shown in Figure 112 on page 114. Networks are typically based on the TCP/IP protocol suite, but the firewall concept as such is not restricted to the TCP/IP protocol suite. Firewalls have become an important concept in TCP/IP-based networks because the global Internet is a TCP/IP-based network and is often perceived as being an unsafe place to enter or traverse. Yet you still want your internal network (perceived as being a safe place) connected to the unsafe Internet.

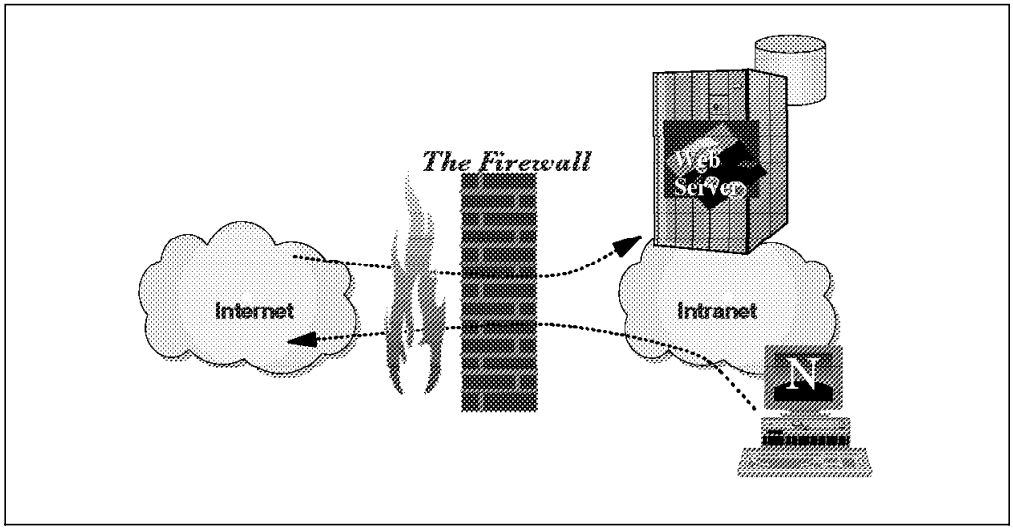


Figure 112. What is a Firewall?

An internal TCP/IP-based network is most often referred to as an *intranet* as opposed to the *Internet* (with a capitalized I) - the world-wide Internet. The term *internet* (with a lower-case i) is a generic term used to refer to an interconnected network of TCP/IP-based networks of which the Internet is an example.

The reasons for establishing connections between intranets and the Internet are many, but generally fall into two categories:

- You want to provide a service to the Internet community.
- You want to allow your internal employees access to the vast amount of services on the Internet and to exchange or share information with other users on the Internet or through the Internet.

A service on the Internet could be as simple as a Web site where you advertise information about your company and your products. Such a Web site can be built and managed very much on a stand-alone basis where the computer that runs the Web server is connected to the Internet, but has no active connections to your internal network. Many Internet Service Providers (ISPs) offer such Web-hosting services. If you out-source your Web site to such an ISP, you can leave the aspects of establishing adequate access security to the ISP, but you are also limited in terms of which types of applications you can offer to the Internet user.

You may already have begun to embark on the path of e-business, where you do business on the Internet in terms of selling your products or interfacing with customers or business partners as part of your daily business processes. Doing e-business on the Internet is very different from just serving static information out of a Web server. Doing e-business means that you have to establish an environment where users on the Internet are able to interact with your “crown jewels,” the applications and data that your daily existence as a company is based on and relies on.

That data and those applications are to a large extent located in your OS/390-based environment, which means that you most likely already are, or in the near term future will be, challenged with the request to establish Internet access to your OS/390 production environment. That is one of the areas where OS/390 Firewall Technologies can assist you.

The other main category of reasons for establishing Internet connectivity is to allow your internal employees access to information and services that are available on the Internet, but without opening a back door through which the “bad guys” on the Internet can gain access to your internal network. This is another area where OS/390 Firewall Technologies can help you.

When you connect your intranet to the Internet and define a strategy for how your firewall should function, you may think that it is sufficient to block all types of traffic that represent a risk and allow the remaining traffic to pass through the firewall. However, such a strategy is based on the assumption that all risks are known in advance and that existing well-behaving traffic will remain well-behaving; such an assumption is a mistake. New ways of exploiting existing applications and well-known application protocols are being found every week, so an application that may be considered harmless today may be the instrument of an attack tomorrow.

5.1.1 General Guidelines for Implementing Firewall Technology

A few general guidelines for implementing firewall technologies are worth including in this context:

- Before you start connecting your internal network to the Internet, make sure that you define a policy for how your firewall should function in cooperation with your security group or security advisors: decide what type of traffic is allowed through the firewall, and under what conditions.
- When actually configuring your firewall, start by disallowing everything and then proceed by enabling those services you have defined in your security policy. Everything that is not specifically allowed is prohibited.
- If you establish more than a single gateway between your internal network and the Internet, make sure that all gateways implement the same level of security. If you build up a perfect firewall on one end of your network while users on the other end dial in to the Internet from their LAN-attached PCs, enabling those PCs to act as IP routers between your internal network and the Internet, a cracker is soon going to exploit that back door into your network instead of wasting his time trying to break through your firewall.
- One of the most important aspects of a firewall is its ability to log both successful and rejected access events. However, these logs are worth nothing if you do not set up daily administrative procedures to analyze and react to the information that can be derived from these logs. By analyzing the firewall logs, you should be able to detect if unauthorized accesses were attempted and if your firewall protection succeeded in rejecting such attacks, or if it failed and allowed an intruder to gain access to resources that should not have been accessed.

This list is not a conclusive list, but merely points out the most important aspects of implementing firewall technologies in your network.

So far, we have only considered the Internet to be the unsafe place, while your internal network has been considered the safe place. However, that may in some situations be an oversimplification. For example, consider a research department that works with highly confidential information. In such an environment, you may want to protect that research department from your regular users by implementing a firewall between your regular internal network and the network in your research department, in this case considering your regular users to be the “bad guys.”

In the following section, we do not refer to the Internet as the generally unsafe network, and an intranet as the generally safe network. We instead use a more generic terminology that is used in most of the firewall literature: the non-secure network and the secure network.

5.2 Firewall Categories

There are many firewall technologies available, but they can in general be grouped into two major categories:

- Those that allow IP packets to be routed between two or more networks
- Those that disable IP routing, but relay data through specialized application programs

5.2.1 Router Firewall

A *router firewall*, as shown in Figure 113, is a machine that routes IP packets, giving the impression of a normal IP router.

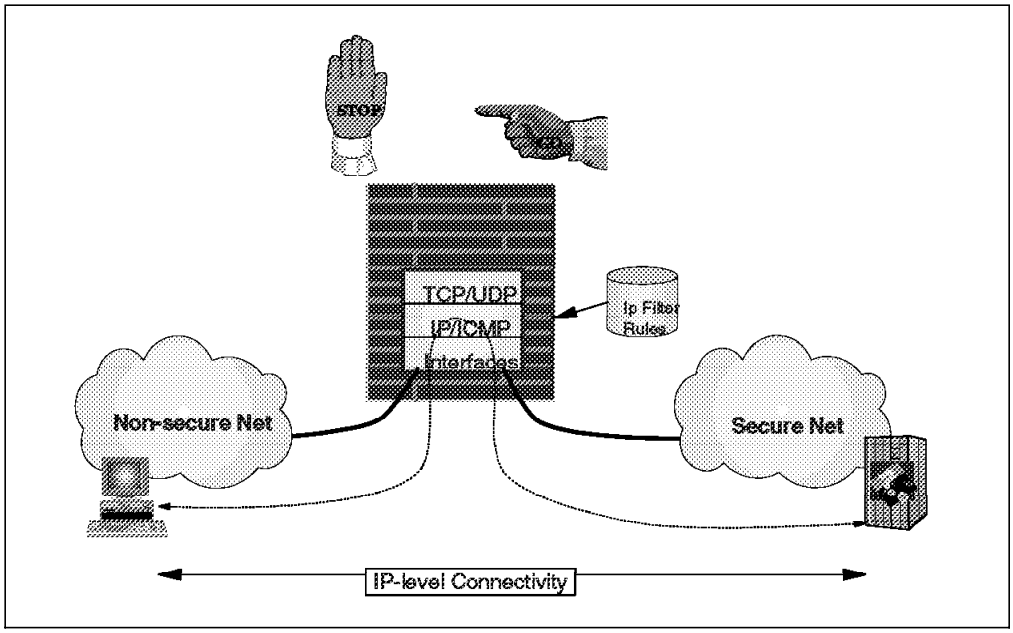


Figure 113. A Router Firewall

What differentiates a router firewall from a normal IP router is that it applies one or more technologies to analyze the IP packets and decide if a packet is allowed to flow through the firewall or not. Such a firewall is sometimes also referred to as a *screening filter*, or *packet filter*. It is worth mentioning that there are many IP routers on the market that implement IP packet filtering technologies, but that does *not* turn these routers into firewalls. They lack all the other characteristics of a firewall, most notably the logging abilities that are so important when dealing with real firewalls. A router firewall may use various technologies to alter or encapsulate the information in the IP packets, such as network address translation (NAT) or tunnelling.

5.2.2 Application Gateway Firewall

An *application gateway firewall*, as shown in Figure 114, is a machine that disables IP-level routing between the non-secure network and the secure network, but allows specialized application gateway programs that run on the firewall to communicate with both the secure network and the non-secure network.

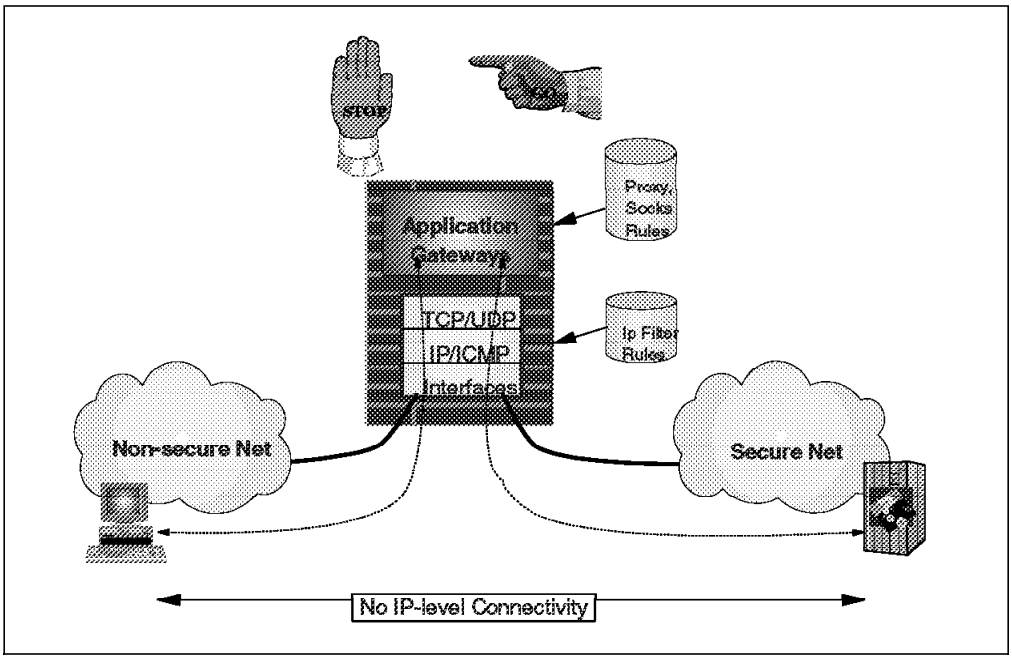


Figure 114. An Application Gateway Firewall

An *application gateway firewall* is sometimes referred to as a *bastion* host. The applications on the firewall act as relay applications between users or applications on the secure and the non-secure networks. Examples of such relay applications are various proxy servers and the socks server.

A firewall may not necessarily have to be configured as either a router firewall or as an application gateway firewall; it may be configured to act as a router firewall for certain application protocols, while it acts as an application gateway firewall for other applications.

5.2.3 Private or Public IP Addresses

One of the problems that the massive growth in Internet use caused a few years back was the threat of using up the available IP address space much faster than originally predicted. This problem was addressed from several angles. One approach was to define a set of IP network addresses as so-called private IP network addresses, as opposed to public IP network addresses. The private address space is defined in *Address Allocation for Private Internets*, RFC1918.

The private address space is made up of the following network addresses:

- 10.0.0.0 - 10.255.255.255 (10/8 prefix) - one class A network
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix) - 16 class B networks
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix) - 256 class C networks

The rationale behind allocating the private address space was that all TCP/IP hosts could be grouped into three groups:

1. Hosts that do not require access to hosts in other enterprises or the Internet. Hosts within this category may use IP addresses that are unambiguous within an enterprise, but may be ambiguous between enterprises.
2. Hosts that need access to a limited set of outside services (such as e-mail, FTP, Web, remote login) that can be handled by mediating gateways (such as an application gateway firewall). For many hosts in this category, unrestricted external access (provided via IP connectivity) may be unnecessary and even undesirable for privacy/security reasons. Just like hosts in the first category, such hosts may use IP addresses that are unambiguous within an enterprise, but may be ambiguous between enterprises.
3. Hosts that need network layer access outside the enterprise (provided via IP connectivity). Hosts in this category require IP addresses that are globally unambiguous.

Globally unambiguous IP addresses are referred to as *public* IP addresses, as opposed to *private* IP addresses.

Private addresses can be used in an intranet as long as the following guidelines are followed:

- IP packets that contain private addresses as source or destination address must be confined to the intranet.

  This means that a host with a private IP address cannot route IP packets to/from hosts outside the intranet. If a host with a private address needs to communicate with hosts outside the intranet, technologies that hide or transform the private addresses have to be in place. The use of application gateway firewall technologies or network address translation technologies meet this criteria.

- Application protocols that carry IP address information such as application data should be used with caution and you should take every possible measure to avoid passing information on private addresses to destinations outside the intranet.

  Application protocols such as the domain name server protocol, the file transfer protocol, or various dynamic route update protocols may carry IP address information in the application data part of an IP packet. Such protocols should be handled with care, and proper configuration of connections between the intranet and the Internet must be established. Special configuration of DNS is required on a firewall to cover this requirement. In general, dynamic routing is not used between an intranet and the Internet, so the dynamic routing update protocols are generally not an issue.

If these guidelines cannot be met, you must apply for public IP addresses for your intranet, which is an activity you would perform together with your Internet service provider.

5.3 The OS/390 Firewall Technology Kit

The OS/390 Firewall Technologies is a kit that you can use in establishing connectivity between your OS/390 system and any network you might consider non-secure, such as the Internet.

By using the kit, you can set up your OS/390 system as a traditional firewall: as a router firewall, or as an application gateway firewall, or as a combination.

You may also use the kit to allow your existing OS/390 production system to play a more active role in your Internet application offerings. By using some of the firewall tools in combination with your standard OS/390 security functions, you may, for example, allow Internet access to a Web server that runs on your OS/390 production system without compromising other applications running on the same OS/390 system.

A good understanding of your requirements, combined with a solid knowledge of the tools you have available, will allow you to define which tools to use and how to use them in order to implement the solution that will meet your requirements.

In this section of the book, we discuss the individual tools that are supplied with OS/390 Firewall Technologies:

- IP filtering
- Network address translation (NAT)
- Virtual private networks (VPN), also referred to as secure IP tunnelling
- Proxy servers
- socks server
- Domain name services

5.4 IP Filtering

IP filters are the most basic building block in a firewall. They are used in all firewall implementations, either on their own or in combination with other firewall tools.

The simplest form of a firewall is a *router firewall*, which acts as a normal IP router between a secure and a non-secure network. By applying IP filter technology along with logging functions to such a router, you can control which packets are allowed to flow in and out of the router firewall.

To configure a router firewall you need to define your filtering rules. Although the concept of applying filter rules to a router may seem simple, the job of defining the correct set of filters is certainly not simple. Filtering rules can become very complex, and to define them requires a thorough understanding of the IP, ICMP, TCP, and UDP protocols.

Before we dig into the details of IP filter rules, note the following words of comfort: OS/390 Firewall Technologies comes preloaded with a broad range of IP filter rules and cluster of filter rules. For most purposes, you will only need to identify which clusters (referred to as services) you need to activate to accomplish what you want. If, for example, you want to allow e-mail to route through the firewall in both directions, you just need to activate the service

called *mail*. This service will then refer to filter rules that are needed to allow this service. OS/390 Firewall Technologies ship with some 25 predefined services and over 100 predefined filter rules.

As shown in Figure 115, IP filtering rules are checked when an IP packet arrives over an interface *and* when an IP packet is sent out over an interface.

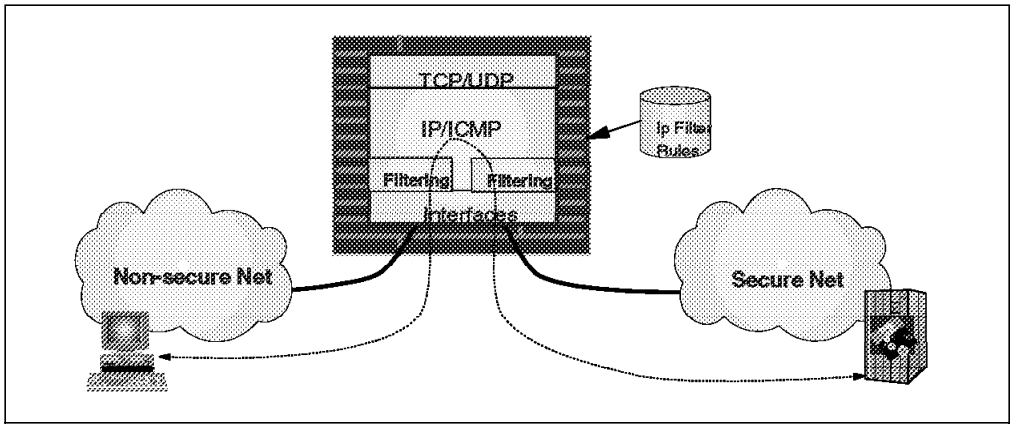


Figure 115. Filter Checking Points

The individual filter rules include attributes that instruct the filtering function whether to permit or deny a packet's continued flow. Provided a previous filter has not denied the packet, a filter rule is always applied. These attributes are based on interface type (secure or non-secure), and direction (inbound or outbound).

As shown in Figure 116, Filter rules are based on information that can be derived from the protocol headers in an IP packet: the IP header, the TCP header, the UDP header, and the ICMP header.

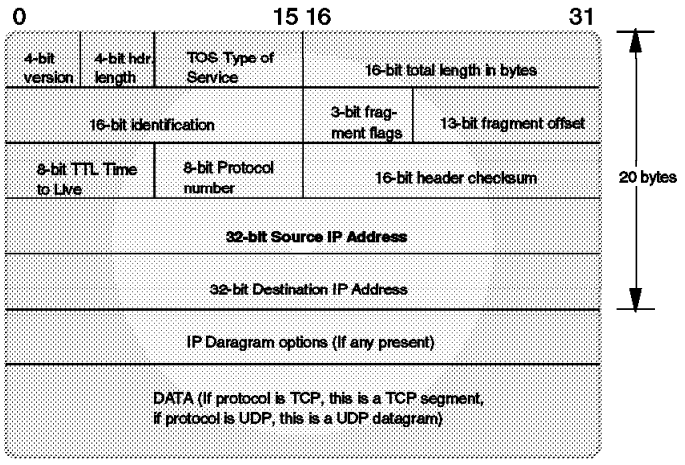


Figure 116. IP Header Fields Used in Filter Rules

The IP header protocol field is used to instruct the receiving IP layer as to what type of data this IP datagram carries. Some of the more common protocols are:

1 Internet Control Message Protocol (ICMP)

4 IP in IP (Encapsulation)

6 Transmission Control Protocol (TCP)

17 User Datagram Protocol (UDP)

50 Secure IP - Encapsulating Security Payload (ESP)

51 Secure IP - Authentication Header (AH)

89 Open Shortest Path First (OSPF)

5.4.1 IP Filter Rule Elements

A filter rule consists of the following basic elements:

Type This attribute specifies what the action of this rule is: to permit the traffic or to deny the traffic.

The remaining attributes of a filter rule are used to identify the traffic that this rule applies to.

Protocol Specifies the type of traffic (in terms of protocol in the IP header this rule applies to:

- **all** All protocols
- **tcp** TCP without an ACK bit. It indicates that this is a TCP connection initiation request. Only the first SYN segment in a TCP segment has no ACK bit - all other segments in a TCP connection will have an ACK bit set. This protocol specification can be used to control which end is allowed to initiate a TCP connection.
- **tcp/ack** Normal TCP segments.
- **udp** UDP datagrams.
- **icmp** ICMP datagrams.
- **ospf** OSPF protocol data units.
- **ipip** Encapsulated IP.
- **esp** Encapsulated secure payload.
- **ah** Authentication Header.

Source op Logic operator to apply to the source port number (any, eq, neq, lt, gt, le, ge).

Source port Source port number. For ICMP protocol, specify the ICMP type.

Dest op Logic operator to apply to the destination port number (any, eq, neq, lt, gt, le, ge).

Dest port Destination port number. For ICMP protocol, specify the ICMP code.

Interface Which interface does this rule apply to: the secure interface, the non-secure interface, or both?

Routing Does this rule only apply to IP packets that originate from or are destined to the firewall machine, or does it apply to IP packets that are supposed to route through this firewall, or to both?

Direction Does this rule apply to IP packets entering or leaving the firewall, or to both?

Logging Specify if a packet that is matched by this rule should be logged or not (yes or no).

Tunnel id If this rule permits the traffic to flow, use this tunnel ID.

Fragment How should this rule be applied in case of IP fragments?

5.4.2 Controlling TCP Connections

TCP is the transport layer protocol that is used by most applications in a TCP/IP-based network. Applications such as FTP, Telnet, SMTP (mail) and HTTP (World Wide Web) are all application protocols that use the TCP transport protocol.

Data is exchanged between two TCP applications over a TCP connection using TCP segments. TCP segments, as shown in Figure 117, are carried in the data part of an IP datagram.

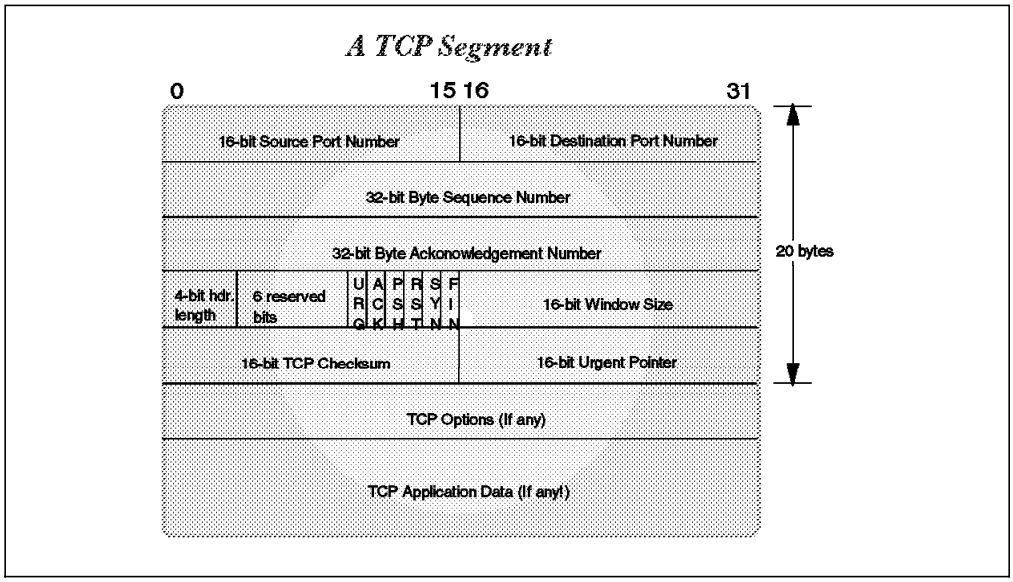


Figure 117. TCP Segment Layout

A TCP connection is initiated by the client that sends a TCP connection request to the server (a SYN segment).

When setting up filter rules, you need the ability to control from where a TCP connection is initiated. You do not want to allow users in the non-secure network to initiate TCP connections to more than the absolute minimum number of services on your gateway. On the other hand, you want your gateway to be able to initiate TCP connections into the non-secure network. To control this, you use information in the TCP header.

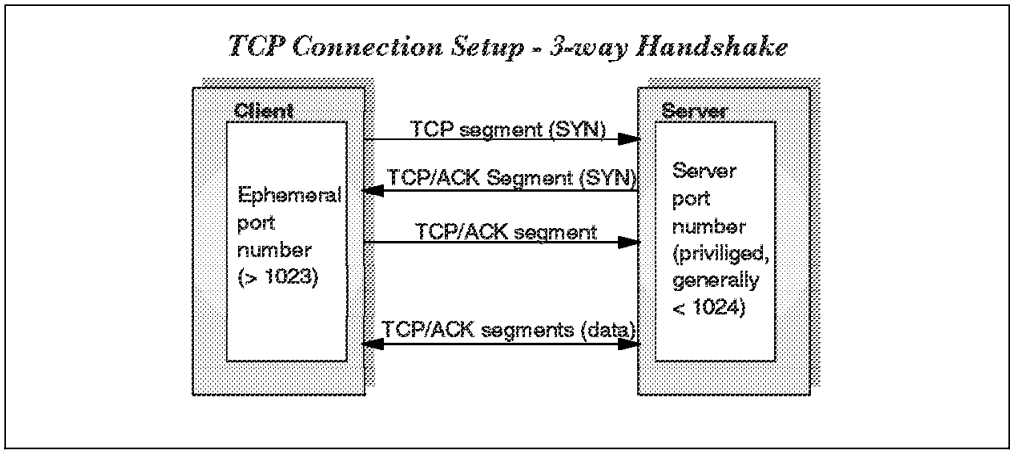


Figure 118. TCP Connection Setup

All TCP segments have an ACK (acknowledgement) bit set, except the very first segment that is sent by the host that initiates the TCP connection. You can use this in your filtering rule. By specifying a protocol of TCP in a filter rule, you include all TCP segments in this rule, with or without an ACK bit. By specifying a protocol of TCP/ACK in a filter rule, you only include TCP segments that have an ACK bit.

To set up rules that will allow host A to initiate TCP connections to host B and exchange TCP segments with each other in both directions, but not allowing host B to initiate a TCP connection to host A, you would need filter rules as follows:

1. Allow A to send TCP protocol packets to B.
2. Allow B to send TCP/ACK protocol packets to A.

The first rule allows A to send the very first SYN segment and all other TCP segments with an ACK bit to B.

The second rule allows B to only send TCP segments that include an ACK bit to A, disallowing B to initiate the connection to A.

The source port and the destination port in the TCP header (see Figure 117 on page 122) are used to identify which process is using a TCP connection. A TCP/IP connection is uniquely defined by:

```
< Source Address, Source Port, Destination Address, Destination Port >
```

To enable clients to connect to servers, servers are most often started on well-known port numbers. Such port numbers are defined for well-known server applications, such as a Telnet server on port 23, an FTP server on port 21 and a Web server on port 80.

Well-known port numbers are generally located below 1024. Servers will therefore generally be located on port numbers below 1024. Clients also have port numbers, but these are assigned by the TCP layer on the client host. A client port number that is assigned by the TCP layer is called an ephemeral (or short-lived) port number and will always be above 1023.

If a service normally uses a well-known port, that does not mean that it cannot use another port. For example, the Telnet server usually uses port 23, but nothing prevents it to be run on another port, for example, port 5234.

This must be considered because it might be used to circumvent the firewall restrictions, either by an outsider or an insider. Often, holes in the firewall security are not directly created by attackers, but by unhappy insiders who consider the firewall to be unnecessarily restrictive. An insider who wants to provide an outside access that is not permitted may use a nonstandard port in order to do it.

For example, if you prevent your users from providing HTTP servers but allow connections from outside to non-privileged ports, a user can provide HTTP access using port 5234.

If an outsider is trying to scan your network including your firewall in order to discover which machines you have and which services you provide, he will use a port scanner. Usually port scanners (such as fwice) try to open a connection to the port. Using the ACK bit checking in the firewall will block the attack.

However, it is possible to scan a network without sending any packet with the SYN bit on. In order to do this, a packet may be sent with the ACK bit on. If the port is active, the host will realize that a connection is not in progress and send a reset response. If the port is not active, there will be no response. Other types of TCP packets may be used to perform similar types of scanning, such as a packet with SYN:FIN, ACK in the header, or one with the flag field set to 0. All of these packets are rejected, but the fact that they are rejected provides some information about the target machine. This is called *stealth scanning*.

If you want to allow IP forwarding on the firewall and rely on the SYN control, you must be aware that your network, including your firewall, might be scanned using these techniques.

5.4.3 Controlling UDP Traffic

UDP, like TCP, is a transport layer protocol, but it is less widely used by applications. Applications using UDP include the domain name service (DNS) and the simple network management protocol (SNMP).

Unlike TCP, UDP does not provide the application with a reliable end-to-end connection. Once a UDP packet has been sent, the sender has no knowledge about whether it has arrived or not. It is therefore up to the application to provide acknowledgment and sequence control, if required. UDP is *connectionless*. That is, each message is a separate entity with no expectation of responses or subsequent request messages. Applications often mimic the operation of a connection-oriented protocol; for example, a client may use a dynamically allocated port to send a message and then listen on that same port for a response.

From the firewall point of view, the only important parts of the UDP packet are the source port and destination port as shown in Figure 119 on page 125 These are used to identify which processes are sending or receiving a UDP datagram.

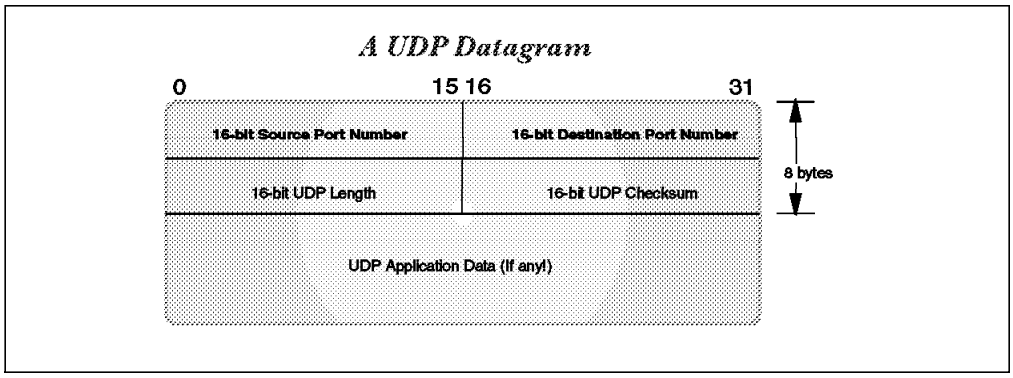


Figure 119. TCP Connection Setup

As in the case of TCP, certain well-known ports are reserved for specific applications. For example, DNS uses port 53, and SNMP uses ports 161 and 162.

Because of its connectionless nature, UDP does not have the three-way handshake sequence of TCP (see Figure 118 on page 123). This means that you do not have the ability to create filter rules based on the direction in which a UDP “session” is established. You should generally avoid routing UDP datagrams through the firewall except between specific, known end points.

5.4.4 ICMP Packets and ICMP Header Fields

ICMP is a protocol designed to communicate errors and information between hosts that are processing IP datagrams. You can find the specification of ICMP in *Internet Control Message Protocol*, RFC792. It is used for purposes such as informing that a host is unreachable or that a sender is sending packets too fast.

Each ICMP message consists of a type plus a code, both of which are small integer values. Unlike the higher layer protocols such as TCP or UDP, there is no source port or destination port, just the message type and code.

When configuring firewall filters, you could disable all ICMP messages in both directions if you do not care about the different types of message. This may make it difficult for you and your users to troubleshoot access problems, but it will be safer and simpler for you. You also have to consider that some ICMP messages are used by network management applications (principally echo request and echo reply).

Typically used ICMP type values are:

0 Echo reply (response to a ping request). Only a code of zero is valid.

Echo request (ICMP type 8) can be used by an outsider to map your network. We suggest you *allow the outgoing echo request and incoming echo reply*. Disable the incoming echo request and outgoing echo reply.

You could consider enabling this facility to some key hosts, such as the router of your network provider. You might allow incoming pings to the non-secure adapter.

3 Destination unreachable notification.

These messages are generated by hosts or intermediate routers in order to notify that a session cannot be established. The ICMP code may lie within the range of 0 to 15 and is used to distinguish between different causes of unreachability. A code value of 1 means that the specified destination IP address cannot be reached. A code value of 3 means that the specified port number in the TCP or UDP header cannot be contacted (an application program does not run on the specified port number).

Outsiders can force nodes of your network to generate these packets in order to obtain knowledge of your network; for example, they can use a port scanner to learn which services you are providing. If you reply with a port unreachable, they then know that you are not providing this service (this type of information can also be gathered for TCP services by using stealth scanning).

You should receive these messages, as they may provide useful information for troubleshooting. You should only send them through the secure interface, because if you send them through the non-secure interface, it will help outsiders map the services that you are offering.

4 Source Quench (optionally sent by an intermediate router if it gets overloaded and begins throwing away IP packets). Only a code of zero is valid.

This message could be used by an attacker (probably combined with IP spoofing) in order to make a very effective denial of service attack. Unfortunately, it is more often a legitimate message, so if you decide to filter it out, you may cause problems due to lost packets. We suggest you

allow it to be sent and received, but also log the received messages for later analysis.

5 ICMP Redirect message.

Used by a directly connected router to instruct the sending host of a better directly connected router for the specified destination. ICMP redirects can be used by crackers to manipulate the routing table of the sending host. The code field is used to distinguish between different types of routes, such as network routes or host routes.

Our recommendation is to send and log this packet, but not to receive it, as your routing tables should be determined only by you. It is also recommended to notify the owners of the machines to which you sent redirects so that they can correct their routing tables.

8 Echo request (ping request). Only a code of zero is valid. (See the preceding description for ICMP type 0 - Echo reply).

9 Router advertisement.

10 Router solicitation. Both type 9 and 10 are used by the ICMP router discovery protocol as described in RFC1256.

We recommend that you do not send or receive any of these two ICMP types on your non-secure interface. Your routing tables should be, as previously mentioned, determined entirely by you and not by ICMP packets received over a non-secure interface.

11 Time to live (TTL) has been exceeded.

Time to live exceeded (code 0) is generated by a router when it has to forward a packet with a TTL value of zero. Fragment reassembly time exceeded (code 1) is generated by a host when it does not receive all the fragments needed to reassemble a packet.

Enable this for incoming packets so your hosts can perform error recovery. For outgoing packets, allow all fragment reassembly time-exceeded messages but not the TTL-exceeded messages.

The reason that we recommend blocking TTL-exceeded messages from going from the secure network to the non-secure network is that an attacker can use a tool called traceroute to find out which hosts are the routers in your network. This tool manipulates the TTL option of a UDP packet in order to receive an ICMP TTL-exceeded message in response. Blocking the outgoing TTL messages will help you hide your network structure.

12 Parameter problem (IP header fields are bad).

This message is generated when a host that is processing a packet finds a problem in the header parameters that forces the packet to be discarded.

An outsider will gain no information with this packet, so allow it to flow in both directions in order to report problems.

13 Time stamp request.

14 Time stamp reply.

this is a simple way of obtaining the time from another host, which is not used very much today. Other upper-level application protocols are used instead, such as DCE time services. There is really no reason to allow it

through the firewall. It could be used by an attacker as an alternative to ping.

17 Address mask request.

18 Address mask reply.

Address mask request and reply can be used by a disk-less system to obtain the subnet mask for a shared media network. It is also not used very much today. Other protocols are used instead, such as BOOTP or DHCP.

This message can be used by outsiders to learn the topology of your network. There were also cases in which a TCP/IP stack took inappropriate actions when it received an unsolicited address mask reply. The address mask request message may be generated by a network management station. Do not allow either message in any direction.

37 Domain name request.

38 Domain name reply.

These messages are used by hosts in order to learn the domain associated with an address. The host sends a domain name request message and receives as an answer a domain name reply. It is specified in *ICMP Domain Name Messages*, RFC1788, and the current status of the protocol is experimental.

The idea of this protocol is to substitute the IN-ADDR domain defined in the domain name server (the one that is used in order to translate IP addresses to domain names). Using this protocol, each host is responsible for the translation of its own IP addresses. The RFC requires every host to implement an ICMP domain name server and also suggests that every host should implement an application for sending the ICMP domain request.

Block it, because it is currently not used.

30 Traceroute.

This message is used in order to implement traceroute (a useful network debugging tool) in a more efficient way. It is specified in RFC 1393, and the current status of the protocol is experimental.

The implementation has two parts:

- A new IP option
- The new ICMP traceroute packet

When a host wants to discover the path to a node, it sends a packet (for example, an ICMP echo request) with the new IP option. Then every router that forwards the packet will also send an ICMP traceroute message to the sender, informing it whether the packet was successfully forwarded or if it was discarded.

If it is incoming (that is, used to trace routes from the secure network to the non-secure network), this packet can be allowed. If you want to hide your internal network structure (and you probably should), the outgoing packet must be blocked.

5.4.5 IP Fragmentation

IP fragments are created if an intermediate router receives larger IP packets on an inbound interface than it is able to send out on an outbound interface.

Maximum packet size varies according to network type. On a Token-ring network, the maximum packet size is around 4K, while on the Ethernet, the maximum packet size is either 1492 or 1500 bytes. If a router receives a 4K IP packet over a Token-ring interface and has to forward it onto an Ethernet interface, it will fragment the 4K IP packet into the necessary number of IP fragments.

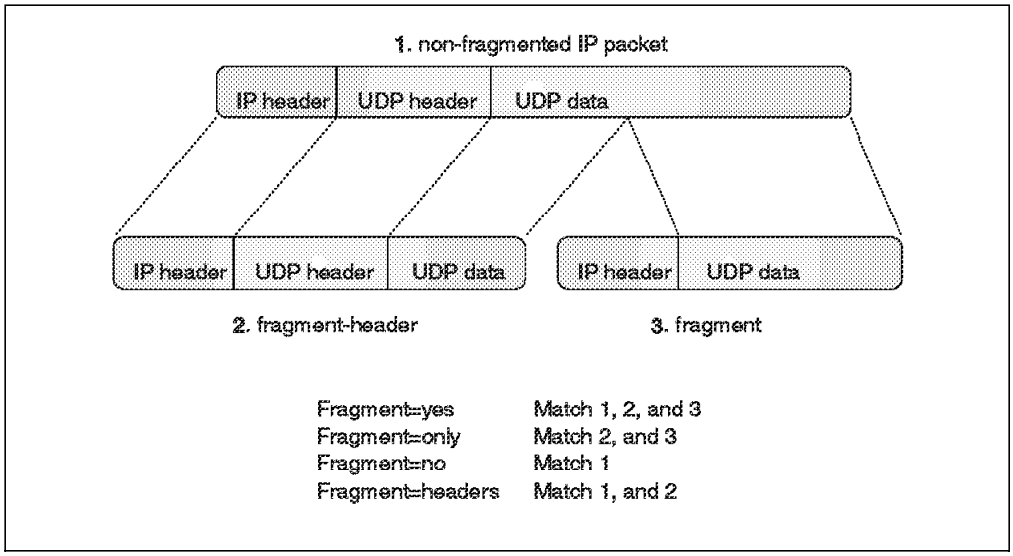


Figure 120. IP Fragmentation

When an IP packet has been fragmented, it will not be reassembled into a full IP packet until it arrives at its final destination. Fields in the IP header are used to control fragmentation and reassembly. If the data in the IP packet is a UDP datagram or a TCP segment, then only the *first* fragment will include the TCP or UDP header, where the port number information is located. The remaining fragments will all have a full IP header, but no TCP or UDP header.

The first fragment, which includes a TCP or UDP header, is called a *fragment header*. You use the fragment attribute on an IP filter rule to control if the rule should be used in checking fragments or not. The possible options are:

yes The rule is used for all IP packets, whether they are fragmented or not. If the IP packet is a fragment, but not a fragment header, the port numbers in the rule are ignored.

only The rule is only used if the IP packet is a fragment. If it is a fragment header, the port numbers are checked. If it is not a fragment header, the port numbers are ignored.

no This rule is only used if the IP packet is not fragmented.

headers This rule is used if the IP packet is not fragmented, or if the IP packet is a fragment header. If it is a fragment, but not a fragment header, the rule is not used.

You can use the fragment attribute to set up different rules for handling non-fragmented IP packets and fragmented IP packets. Some known attacks are based on misuse of the fragment control fields in the IP header, and in some

cases you might want to permit non-fragmented IP packets to pass, while you deny fragmented IP packets from passing your filter.

The IP specifications allow packets of very small sizes. The minimum packet size that can be sent according to RFC 791 is 68 bytes. The problem here is that this packet size is not enough in all situations to carry the complete information for upper-layer protocols. This leads to an attack technique called the *tiny fragment attack*.

The reassembly algorithm contains a mechanism by which later fragments can overwrite the data portions of previous fragments. An attacker could create a series of packets in which the first fragment is allowed by the filter, but later fragments overwrite relevant information (such as TCP source and destination ports). In this way the filtering rules can be bypassed if you allow fragmented packets. This is called the *overlapping fragment attack*.

You should consider configuring your firewall to only support non-fragmented packets. See *Security Considerations for IP Fragment Filtering*, RFC1858 for a complete discussion about this point.

5.4.6 When to Use IP Filter Rules

Filter rules are checked *from top to bottom*. Therefore, the order of your filter rules is important. Always place your most specific rules at the top, and your general rules at the bottom. The firewall always adds a default rule at the bottom that is used to deny all traffic over all interfaces.

It is obvious that IP filters are used when you configure a firewall as a router firewall. What may not be so obvious is that you also use IP filters when you configure your firewall as an application gateway firewall. In the case of an application gateway firewall, you use the filters to control which requests are allowed to enter the firewall on which interfaces and pass up to the application layer on the firewall and pass out from the application layer to the network interfaces.

If, for example, you want to run a socks server on your firewall and allow users in your secure network to go through the socks server and into the non-secure network, you would configure IP filter rules that allowed local, inbound connection requests on the secure adapter for the socks server port number (1080). You would also need to set up filter rules that would allow the socks server to return TCP segments to the client in the secure network. See Figure 121 on page 130 for the two rules we just described. (You would actually need a few more IP filter rules to accomplish the full implementation for socks.)

```
          1                          2
       type = permit              type = permit
   protocol = tcp             protocol = tcp/ack
 srcopcode = gt             srcopcode = eq
   srcport = 1023             srcport = 1080
destopcode = eq           destopcode = gt
  destport = 1080            destport = 1023
 interface = secure         interface = secure
   routing = local            routing = local
 direction = inbound        direction = outbound
       log = no                   log = no
    tunnel =                   tunnel =
  fragment = yes             fragment = yes
```

Figure 121. Sample IP Filter Rules

1 The first rule allows clients in the secure network to initiate TCP connections over the secure interface to the socks server on port 1080. In addition, it allows these clients to send normal IP packets that include an ACK segment.

2 The second rule allows the socks server to send IP packets to the clients in the secure network; but because the protocol is specified as tcp/ack, it disallows the socks server from initiating new TCP connections into the secure network.

5.5 Network Address Translation (NAT)

The purpose of NAT is to hide the IP address information in one network from another network. You can do so by using an application gateway firewall, but if you need access to application protocols for which there is no application gateway implementation, an application gateway firewall may not be a viable option.

Consider, for example, that you have an internal network that is based on the private IP address space, and you want to use an application protocol for which there is no application gateway; your only option is to establish IP-level connectivity between hosts in your internal network and hosts on the Internet. You cannot send IP packets with private IP addresses as source IP address through a router into the Internet, because the routers in the Internet would not know how to route IP packets back to a private IP address. As shown in Figure 122 on page 131, the way NAT handles this is by translating the private IP addresses in outgoing IP packets to public IP addresses, and translating public IP addresses to private IP addresses for incoming IP packets.

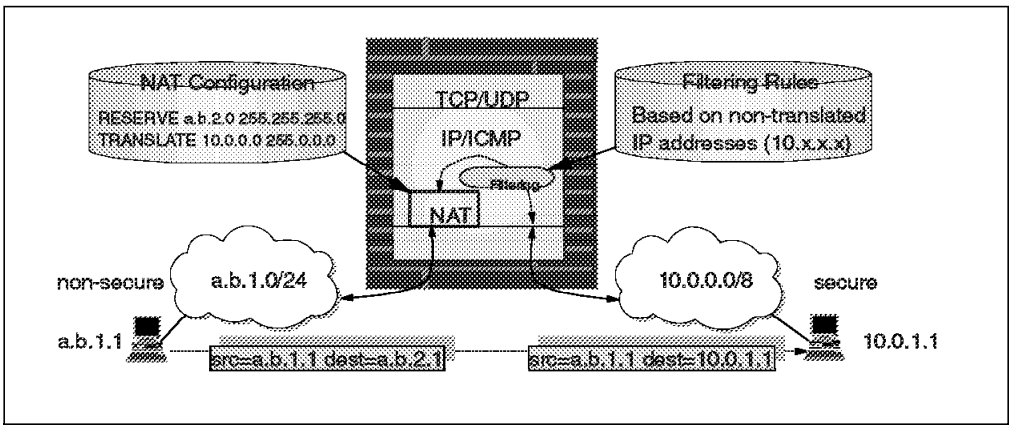


Figure 122. Network Address Translation (NAT)

Seen from the two hosts that exchange IP packets with each other, one in the secure network and one in the non-secure network, NAT looks like a normal IP router that forwards IP packets between two network interfaces (see Figure 123).

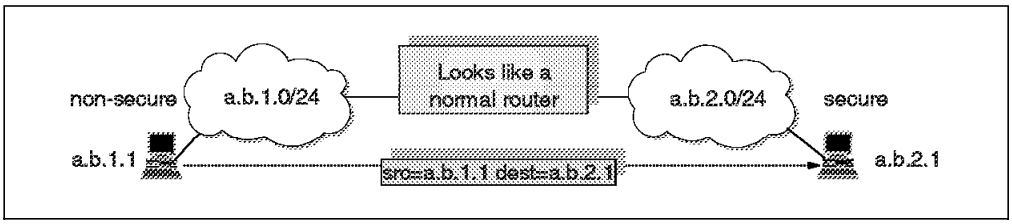


Figure 123. Network Address Translation Seen from the Non-Secure Network

Another area where NAT has proven to be useful is in solving IP address ambiguity in situations where, for example, two companies merge into one company and the IP networks of the companies need to be integrated into each other.

To use NAT, your OS/390 TCP/IP stack must have IP forwarding enabled; it must be able to route IP packets between the interfaces.

NAT will only translate IP addresses in IP packets that carry TCP segments or UDP datagrams. IP packets with ICMP datagrams will not be processed by NAT. You need to take this into consideration when setting up your IP filter rules, so you only allow TCP and UDP to route through your NAT function. You can still allow selected ICMP packets to be received or sent over the non-secure interface by permitting selected local ICMP traffic (routing=local). You need to add filter rules that prevent ICMP packets from being routed through your firewall (routing=route).

Because the TCP/IP stack that implements NAT looks like a normal IP router, you need to create an appropriate IP network design for connecting two or more IP nets or subnets through a router. The NAT IP addresses need to come from separate nets or subnets, and the addresses need to be unambiguous with respect to other nets or subnets in the non-secure network. If the non-secure network is the Internet, the NAT addresses need to come from a public net or subnet. In other words, the NAT addresses need to be assigned to you by your Internet Service Provider.

When you configure NAT, you need to reserve a pool of non-secure IP addresses that are available for use by NAT. You define those addresses to NAT via a NAT RESERVE configuration command.

If connections are established from the secure network, NAT can just pick the next free public address in the NAT pool and assign that to the requesting secure host. NAT keeps track of which secure IP addresses are mapped to which non-secure IP addresses at any given point in time, so it will be able to map a response it receives from the non-secure network into the corresponding secure IP address. When NAT assigns IP addresses on a demand basis, it needs to know when to return the non-secure IP address to the pool of available IP addresses. There is no connection setup or tear-down at the IP level, so there is nothing in the IP protocol itself that NAT can use to determine when an association between a secure IP address and a NAT non-secure IP address is no longer needed. You have to configure a timeout value that instructs NAT how long to keep an association in an idle state before returning the non-secure IP address to the free NAT pool. The default for this parameter is 15 minutes, but may be configured as high as 240 minutes. Keep in mind that the larger the value you set, the larger the pool of NAT non-secure IP addresses you need.

You also need to instruct NAT whether all your secure hosts are allowed to use NAT or not. You do that via NAT TRANSLATE and EXCLUDE configuration commands. The TRANSLATE command instructs NAT which secure addresses should be allowed to use NAT. The command can refer to a range of addresses or just a single address. If it refers to a range, you may want to exclude some addresses from that range, for which you can use the EXCLUDE command.

If hosts in the non-secure network need to initiate connections to hosts in the secure network, you need to tell NAT in advance which non-secure NAT address matches which secure IP address. You define such static mapping via a NAT MAP configuration command. Your external name server may, for example, have an entry for a mail gateway that runs on a computer in your secure network. The external name server resolves the public host name of your mail gateway to a non-secure IP address that is in your NAT net or subnet(s), and the remote mail server sends a connection request to this IP address. When that request comes to NAT on the non-secure interface, NAT looks into its mapping rules to see if it has a static mapping between the specified non-secure public IP address and a secure IP address. If so, it translates the IP address and forwards the IP packet into the secure network to your mail gateway.

The non-secure NAT addresses you assign as statically mapped to secure IP addresses should not overlap with the addresses you specify as belonging to the pool of non-secure addresses NAT can use on a demand basis.

NAT works fine for IP addresses in the IP header. Some application protocols exchange IP address information in the application data part of an IP packet, and NAT will generally not be able to handle translation of IP addresses in the application protocol. OS/390 Firewall Technologies handles one case where an IP address is used in the application protocol, and that is for the FTP protocol.

The FTP protocol includes a PORT command where one end of an FTP control connection instructs the other end on which port number and IP address to open an FTP data connection for transfer of a file. This PORT command is part of the FTP protocol, which from an IP point of view is application data. NAT on OS/390 does scan for an FTP PORT command and translates the IP address in an FTP

PORT command to match the non-secure IP address NAT has chosen for the secure IP address in question.

5.6 Virtual Private Network (VPN)

Consider an example where you have two secure networks you want to interconnect. If the two networks are adjacent to each other, you just set up an IP router between them and traffic will flow freely between the two networks. If the two networks are located far from each other, for example, on two different continents, you might want to use an intermediate non-secure network, such as the Internet, to transport data between the two secure networks.

However, if you use an intermediate non-secure network to transport your secure data over, you probably want to make sure that no one in the non-secure network can eavesdrop on your secure data and that no one can modify the data as it passes through the non-secure network.

To accomplish these two objectives, you can use OS/390 Firewall Technologies to set up and operate a secure tunnel through a non-secure network. Such a secure tunnel is often referred to as a virtual private network (VPN) (see Figure 124).

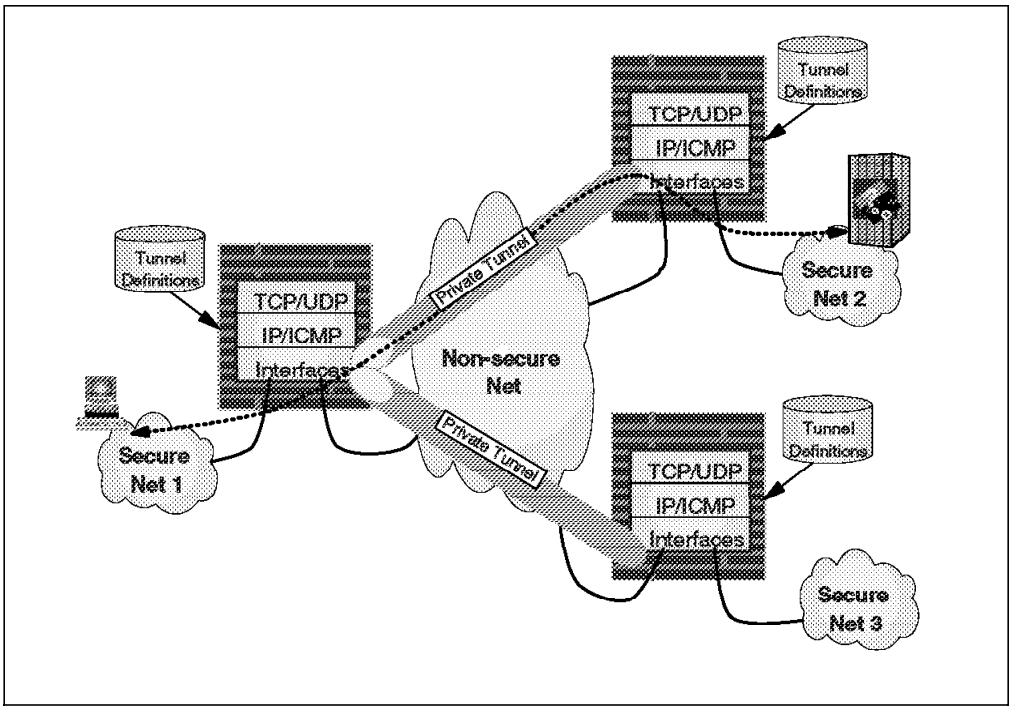


Figure 124. Virtual Private Networks

Secure tunnels in OS/390 Firewall Technologies are based on the IPSec specifications as documented in the following three RFCs:

- *Security Architecture for the Internet Protocol*, RFC1825
- *IP Authentication Header*, RFC1826
- *IP Encapsulating Security Payload (ESP)*, RFC1827

These RFCs describe numerous standards and use of the standards. OS/390 Firewall Technologies implements what is known as *manual* tunnels, where the two tunnel partners (the two tunnel end points) manually exchange the

information that is needed in order to establish a tunnel. This information includes identification information, which cryptographic algorithms to use for authentication and/or encryption, and the keys used by those algorithms.

The tunnel partner can be another OS/390 system, an AIX/6000 system, or any system that implements manual tunnels based on the previously mentioned RFC specifications.

Note: In the first release of OS/390 Firewall Technologies there is no support for dynamic tunnels. You would need that support to establish a tunnel with a workstation running, for example, the Windows 95 Secure Remote Client.

For each manual tunnel you define, you have to specify a *tunnel policy*. A tunnel policy specifies if the data that is sent over the tunnel should be authenticated or encrypted or both.

When working with cryptographic technologies, you generally aim at one or more of the following objectives:

- Authentication

  Knowing that the data received is the same as the data that was sent and that the claimed sender is in fact the actual sender.

- Integrity

  Ensuring that data is transmitted from source to destination without undetected alteration.

- Confidentiality

  Communicating such that the intended recipients know what was being sent but unintended parties cannot determine what was sent.

- Non-repudiation

  Ensuring that a receiver is able to prove that the sender of some data did in fact send the data even though the sender might later desire to deny ever having sent the data.

  IPSec does not ensure non-repudiation.

The IP authentication header standard, as described in RFC1826, is intended to provide integrity and authentication without confidentiality. If you use the authentication header standard, an authentication header is constructed by the sender and inserted into the IP datagram between the IP header and the upper-level protocol data unit that is transported in the IP datagram, such as TCP, UDP, or ICMP. The protocol field in the IP header is set to 51, as shown in Figure 125 on page 135, which indicates the presence of an authentication header. The authentication header itself includes a field that instructs the receiver what the real upper-level protocol is.

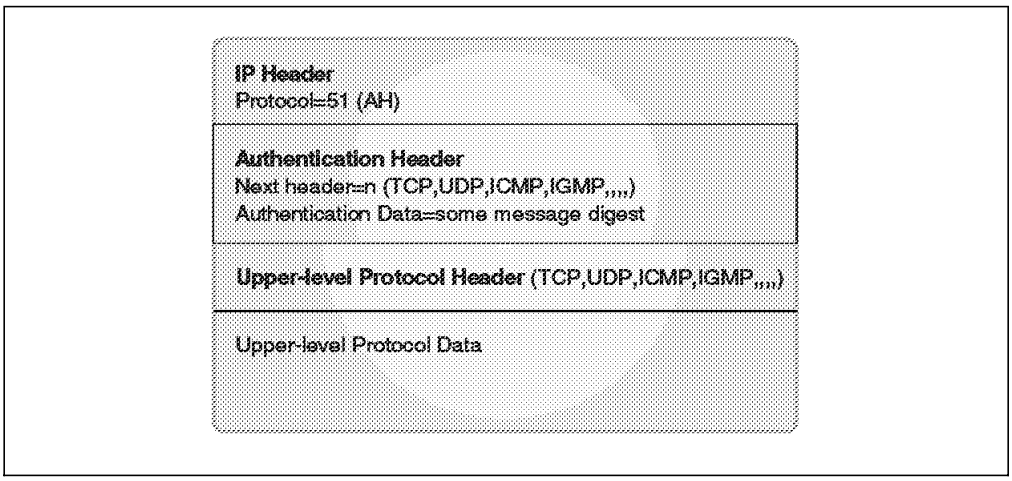


Figure 125. IP Datagram with Authentication Header

In the case of OS/390, a keyed MD5 algorithm is used to calculate an authentication data field that is part of the authentication header. The authentication data field is calculated based on the total IP datagram contents, substituting fields that are allowed to change in transit with binary zeroes, such as the TTL IP header field. The authentication data field itself is also replaced with binary zeroes during the calculation because it is not added to the IP datagram until after the calculation has completed. The IP datagram itself is not encrypted when you use the authentication header standard, but is transported in clear text from sender to receiver.

If you need authentication and integrity only and confidentiality is not a concern to you, you only need the authentication header support as described so far.

If, in additional to authentication and integrity, you also need confidentiality, then you need the support that is provided by the encapsulating security payload standard as documented in RFC1827.

As the name suggests, encapsulating security payload takes a protocol data unit, encrypts it, and encapsulates it into an IP datagram. Two modes are supported by the encapsulating security payload: *transport mode* and *tunnel mode*. OS/390 Firewall Technologies uses the tunnel mode.

In tunnel mode, the encapsulating security payload will take a full IP datagram, encrypt it, and encapsulate it into another IP datagram with an ESP header, as shown in Figure 126 on page 136

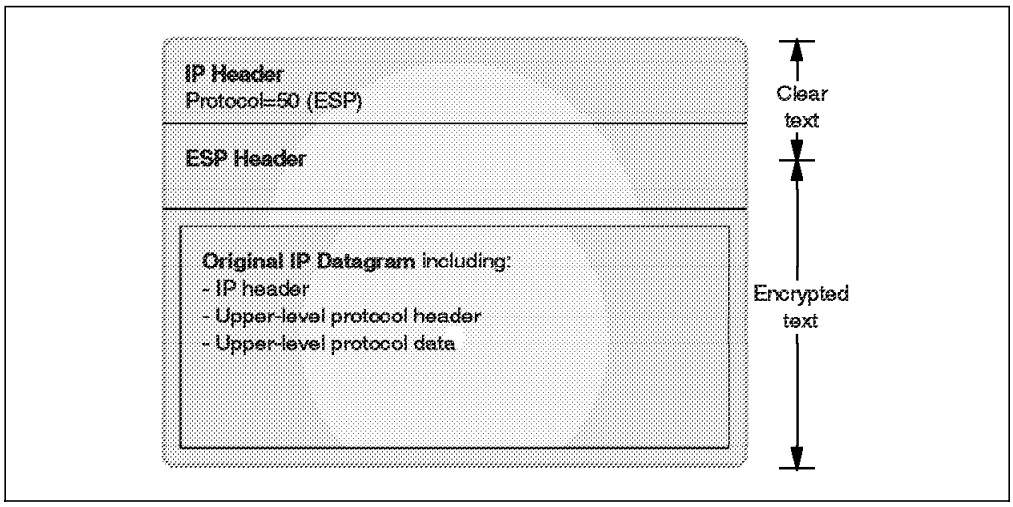


Figure 126. Encapsulating Security Payload in Tunnel Mode

The authentication header standard and the encapsulating security payload standard may be used on their own, or they may be combined. When you set up a tunnel definition, you define the tunnel policy to be used for this tunnel, as follows:

auth Use only the authentication header standard.

encr Use only the encapsulating security payload standard.

ae Use both standards. Create an authentication header after the IP datagram has been encrypted. If you want the receiver to be able to authenticate the full IP datagram, including the encapsulating IP header, use this option.

be Use both standards. Create an authentication header before the IP datagram has been encrypted. If you are only concerned with the receiver being able to authenticate the encapsulated IP datagram, use this option. Be aware that the encapsulating IP header is not authenticated if you use this option.

Figure 127 on page 137 shows both authorization after and authorization before examples.

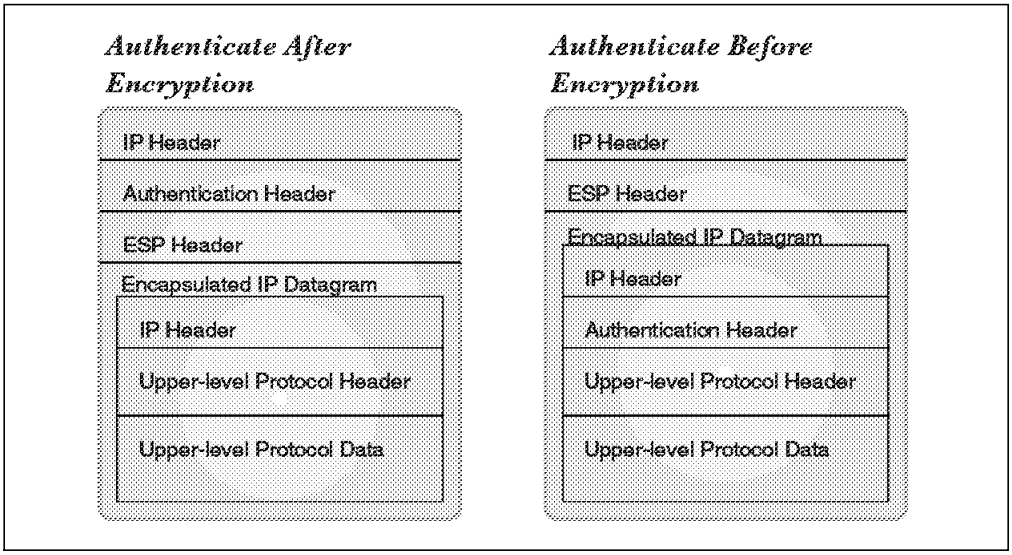


Figure 127. Authenticate after or before Encryption

A tunnel is defined in terms of a tunnel owner and a tunnel partner. The tunnel owner creates all the tunnel definitions and exports them to the tunnel partner, who then has to import them (see Figure 128). How to actually transport the exported tunnel definitions from the tunnel owner to the tunnel partner is not defined for manual tunnels. The definitions may be shipped by courier on a diskette, or transmitted over the network encrypted by some other application protocol, or even carried by hand.

When the tunnel owner defines a new tunnel, the home address of the tunnel owner is entered along with the IP address of the tunnel partner. The security parameter index of the partner has to be entered when the tunnel is defined. The value should be agreed on with the partner before defining the tunnel. If the partner has more tunnels defined, some security parameter index values may already be in use. The tunnel owner's own security parameter index value is generated by the tunnel definition utility.

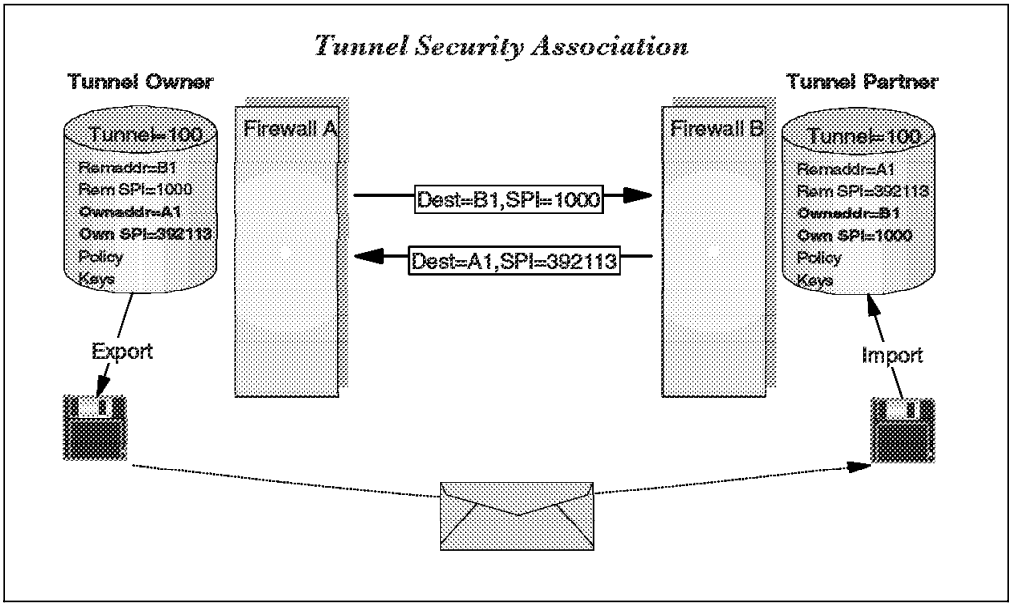


Figure 128. Tunnel Owner and Tunnel Partner Overview

A sending host selects a tunnel definition based on your IP filtering rules. The tunnel definitions are then used to apply the agreed upon processing to the IP datagram before sending it out onto the non-secure network. The security parameter index of the destination host is sent along in the authentication header or the encapsulating security payload header.

When an IP datagram that has an authentication header or an encapsulating security payload header arrives at a destination host, this host uses the security parameter index value along with the destination IP address in the IP packet (its own IP address) to find a matching tunnel definition among its tunnel definitions. The tunnel definition has all the information the receiver needs to process the IP packet according to the policies established between the two tunnel partners.

When you use the encapsulating security payload standard, the encryption algorithms you may use are restricted depending on your home country. In the USA and Canada, DES may be used. Other countries are generally only allowed to use the Commercial Data Masking Facility (CDMF).

There are no such limitations for the authentication algorithm. When using authentication, you do not actually encrypt the data you transmit, you just use an algorithm to compute a message digest. The only algorithm that is supported by the OS/390 Firewall Technologies product is keyed MD5.

For both encryption and authentication, the algorithms are based on symmetric keys. When you create a tunnel definition, the necessary keys are generated and carried over to the tunnel partner in the exported tunnel definitions. You need to handle the exported tunnel definitions with care. If they fall into the wrong hands, you might compromise your tunnel-based security.

5.7 Proxy Applications

A *proxy server* is an application-specific relay server that runs on the host that connects a secure and a non-secure network. The purpose of a proxy server is to control exchange of data between the two networks at an application level instead of an IP level. By using a proxy server, you can disable IP routing between the secure and the non-secure network for the application protocol your proxy server is able to handle, while still able to exchange data between the networks by relaying it in the proxy server. Figure 129 on page 139 shows an FTP proxy server.

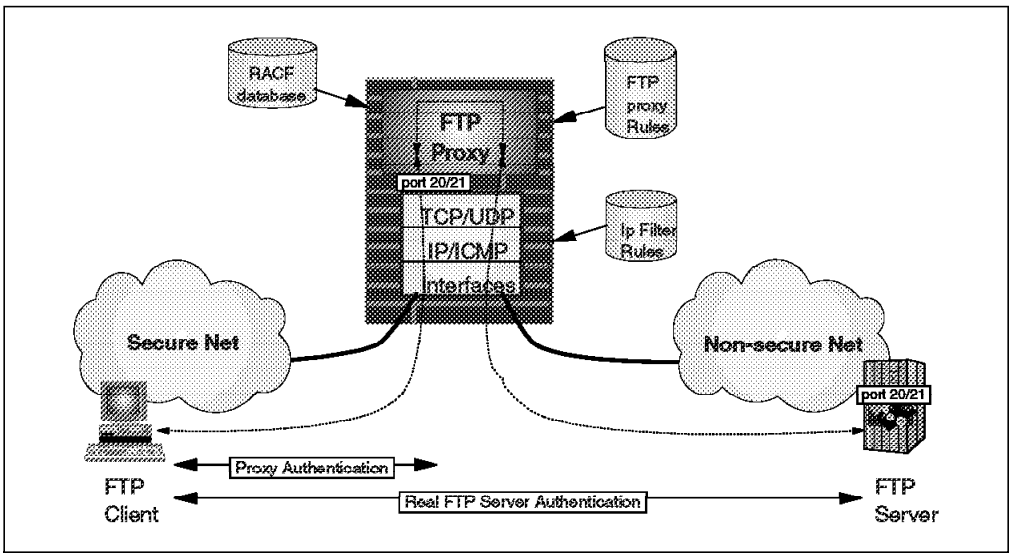


Figure 129. FTP Proxy Server

A proxy server is application-specific, which means it is written for a specific application protocol in mind, such as an FTP proxy, a Web proxy, or a Telnet proxy. Because a proxy is application protocol-aware, it is able to actively participate in the application protocol and, for example, request that the client end user enter a valid user ID and password for the machine on which the proxy executes. Only if the client end user authenticates successfully on the proxy machine is the client allowed to pass through to the other network.

The proxy fully understands the application protocol for which it is written, which means that both the client code and the real server code are unaware of the proxy's presence; no modification to either client or server code is required to use a proxy. The end user that uses the client may have to answer a few extra prompts and type in some extra information to go through the proxy, but the proxy will work with all standard clients written for the specific application protocol. Figure 130 shows a proxy server TCP segment flow example.

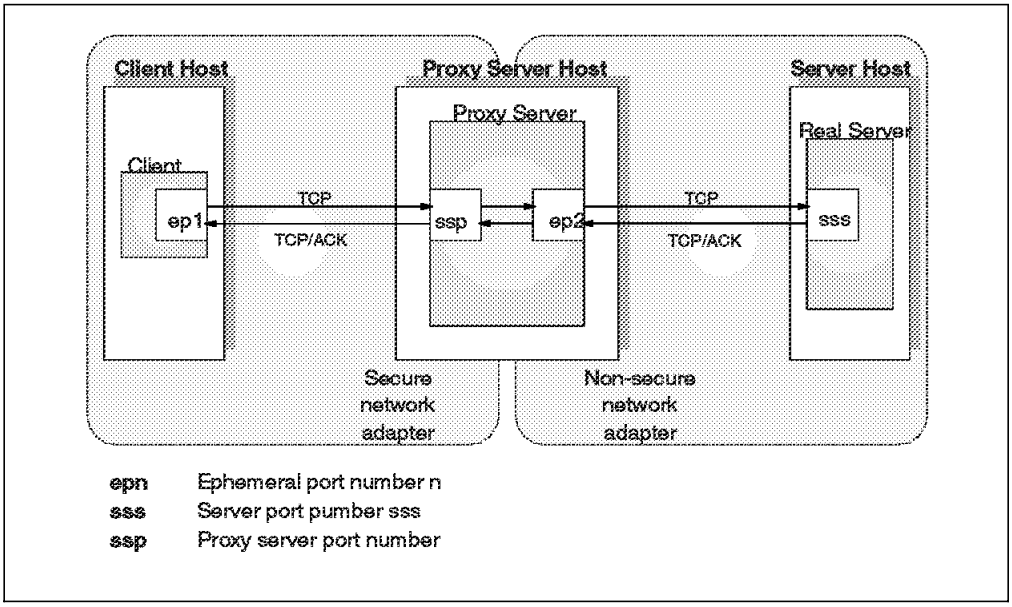


Figure 130. Proxy Server TCP Segment Flow

In order to allow a proxy server to operate on your OS/390 system, you need to add filter rules that allow hosts in the secure network to establish connections to the proxy server port number, and filter rules that allow an ephemeral port number on the proxy server host to establish connections to the real server port number in the non-secure network.

5.7.1 FTP Proxy Server

Important

OS/390 Firewall Technologies includes an FTP proxy that can be used by FTP clients both in the secure and in the non-secure network. If it is used by clients in the non-secure network, be aware that user IDs and passwords flow over the non-secure network in clear and can be traced by any intermediate router. We strongly discourage you from using the FTP proxy from clients in the non-secure network.

In order to use the FTP proxy server on OS/390, users must have a valid OS/390 user ID and password. In addition, users of the OS/390 FTP proxy server must be defined as users of OS/390 UNIX System Services with a valid UID, GID, and home directory.

FTP can be used in one of two modes:

1. Normal mode
2. Passive mode

In normal mode, the FTP client first connects to the FTP server port 21 to establish a control connection. When data transfer is required (for example, as the result of a dir, get, or put command), the client sends a PORT command to the server instructing the server to establish a data connection from the server′s data port (port 20) to a specified ephemeral port number on the client host.

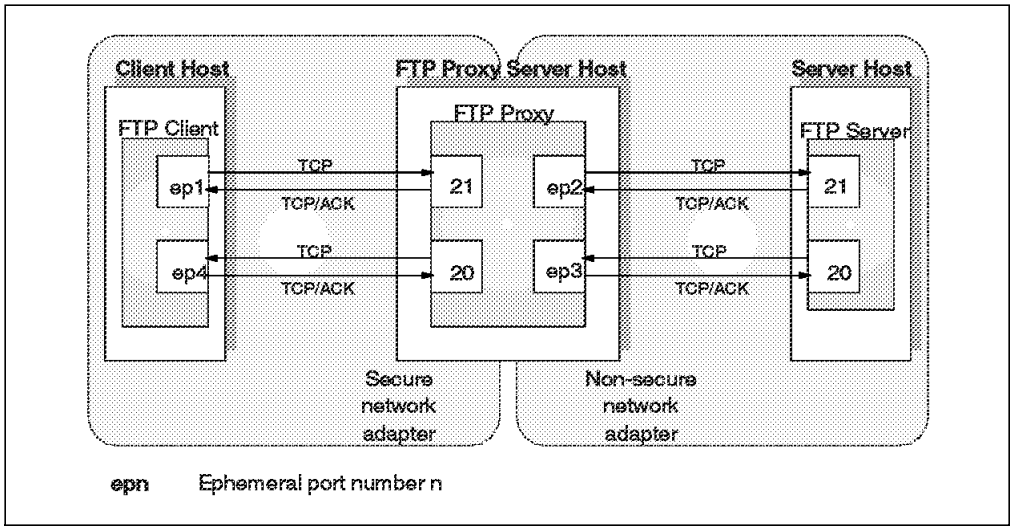


Figure 131. Normal Mode FTP Proxy

In an FTP proxy server situation, *normal mode* means that we have to allow inbound TCP connections from the non-secure network to the FTP proxy host. Notice in Figure 131 how a connection is established from the FTP server port 20 in the non-secure network to the FTP proxy server′s ephemeral port number. To

allow this to happen, you need IP filtering rules that allow inbound connection requests from port 20 to an ephemeral port number on the FTP proxy host. This is normally not an IP filter rule you would want to add to your filter rule configuration, because it would allow a cracker to run a program on port 20 and scan all your port numbers above 1023, which in its simplest form might result in a denial of service situation.

A much more firewall-friendly mode is the passive mode of operation, as shown in Figure 132. This mode has actually been dubbed firewall-friendly FTP, and is described in *Firewall-Friendly FTP*, RFC1579.

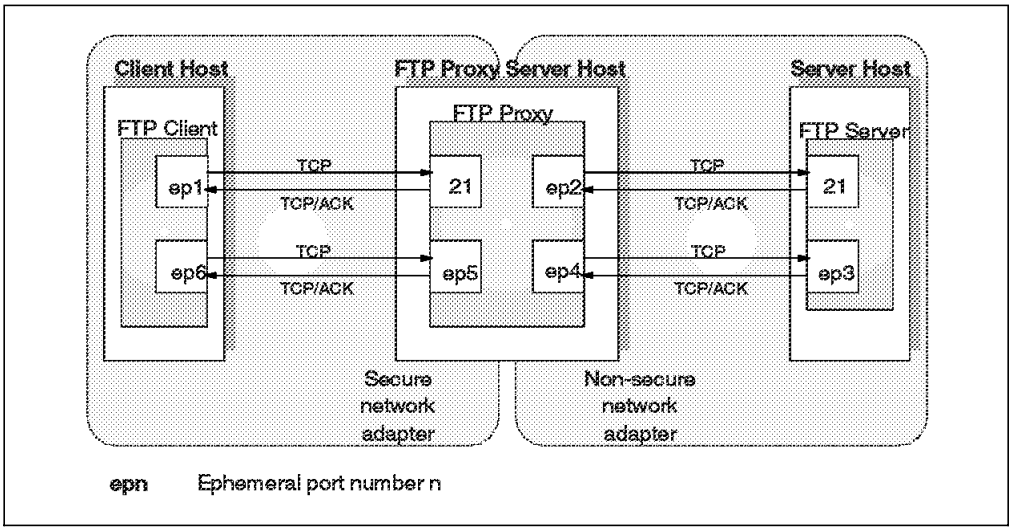


Figure 132. Passive Mode FTP Proxy (Firewall-Friendly FTP)

In passive mode the FTP client again establishes a control connection to the server's port 21. When data transfer has to start, the client sends a PASV command to the server. The server responds with a port number for the client to contact in order to establish the data connection, and the client then initiates the data connection.

In this setup, we need to allow an ephemeral port number on our FTP proxy host to establish connections to both port 21 and any ephemeral port number in the non-secure network. But we avoid adding a rule that allows inbound connections to ephemeral port numbers on our proxy FTP server host.

5.7.2 Other Proxy Servers on OS/390

OS/390 Firewall Technologies does not, in the initial release, include other proxy servers. A Lotus Domino Go Web Server can be configured as a proxy Web server. In such a configuration, you can use a Domino Go Web Server proxy to allow users in your secure network to access Web servers in the non-secure network. The advantage of using a proxy Web server instead of a socks server is that you can authenticate your secure users on the proxy Web server in terms of valid OS/390 user IDs and passwords, before they are allowed to access the non-secure network.

5.8 The Socks Server

The basic principles of a socks server are very much like those of a proxy server: it acts as a relay application on the socks server host where it relays application data from one network to another (see Figure 133). Where the proxy server was application protocol-specific, a socks server is application protocol-independent, and will in general work with all application protocols.

The socks server that is implemented in the first release of OS/390 Firewall Technologies is based on the socks Version 4 specification, which means that it works with TCP-based applications, but not UDP applications. This limitation is normally not considered severe because the vast majority of application protocols are based on TCP. A newer version of socks has been defined, called socks Version 5. This new version includes, among other improvements, support for UDP applications.

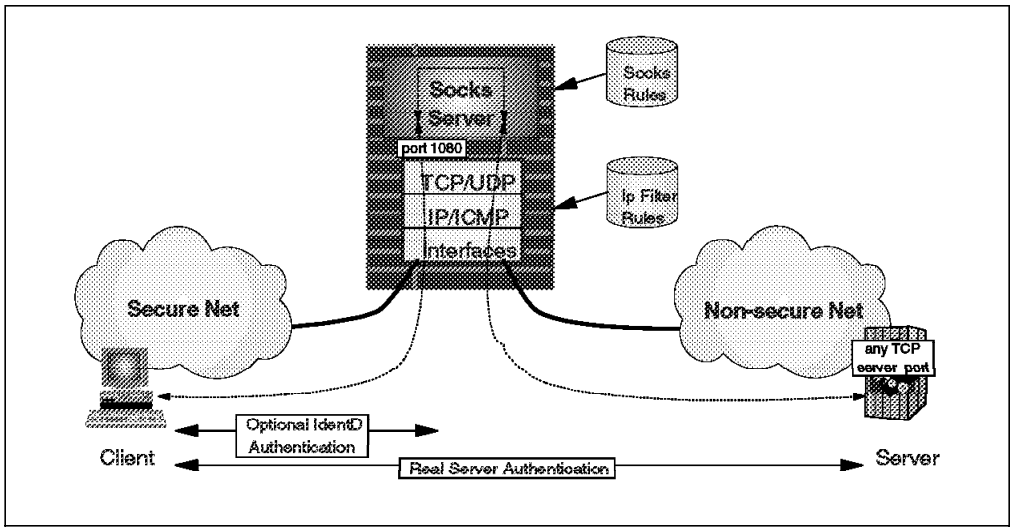


Figure 133. Socks Server

Clients in the secure network connect to the socks server on port 1080. Because socks is application protocol-independent, it does not know how to use the application protocol to add any authentication functions on the socks server host. A socks server makes decisions on whether a connection should be allowed or not on the basis of two tools:

1. A socks server configuration file that instructs the socks server which clients to allow access to the socks server and for which application protocols (expressed as server port numbers).
2. A socks server may optionally (based on your configuration of the socks server) use a protocol, called the Ident protocol, to authenticate a client user before allowing the client end user to use the socks server.

The Ident protocol is based on *Identification Protocol*, RFC1413. This protocol relies on an IdentD server running on the client host on port 113. If the socks server uses IdentD, it connects to the IdentD server on the client host and verifies the information the client program sent to the socks server in the initial data segments. If IdentD authenticates this information successfully, socks accepts the client requests and sets up the TCP connection to the real server in the non-secure network.

You configure how socks should work with respect to the Ident protocol, as follows:

- Do not use the Ident protocol at all.
- Try to connect to an IdentD server on the client host. If an IdentD server is active, then verify the user and allow the connection only if verification is successful. If there is no IdentD server on the client host, then allow the connection. This mode of operation is established by starting socks with a -i flag.
- Try to connect to an IdentD server on the client host. If an IdentD server is active, then verify the user and allow the connection only if verification is successful. If there is no IdentD server on the client host, then reject the connection. This mode of operation is established by starting socks with a -I flag.

An issue with the use of socks is that clients must be made aware that they communicate through a socks server. The connection setup is done in two steps:

1. The client first establishes a TCP connection to the socks server host on port number 1080 and not the real server IP address and port number that the client actually wishes to establish a connection to.
2. The client then sends information to the socks server about what IP address and port number in the non-secure network it really wishes to connect to. If the socks server accepts the request based on socks filter rules and optional use of the Ident protocol, it then establishes a connection to the real server in the non-secure network. From then on socks just relays data segments between the secure network and the non-secure network.

A client that is socks-aware is called a *socksified* client. A client program can be socksified in one of three ways:

- The application programmer adds application code to support the initial handshaking with the socks server.
- The socket library that the programmer uses to compile and link the socket program is a socksified socket library, and automatically adds functions to support connections through a socks server.
- The TCP/IP protocol stack is in itself socksified, and you just configure your TCP/IP protocol stack to use socks whenever needed.

It is obvious that the first choice is the least desirable. The last two options remove the burden of implementing the socks protocol from the application programmer, and change the decision whether or not to use socks into a configuration question. This configuration is generally done by specifying which destinations should be contacted directly (without using the socks protocol), and which destinations should be contacted through a specified socks server host. On most operating system platforms that use either socksified socket libraries or socksified TCP/IP stacks, this configuration is done in a socks.conf or socks.cnf file in the /etc directory. The entries in this file specify address ranges for which not to use socks, and the address of a socks server for all other IP addresses. Figure 134 on page 144 shoes a socks TCP segment flow example.

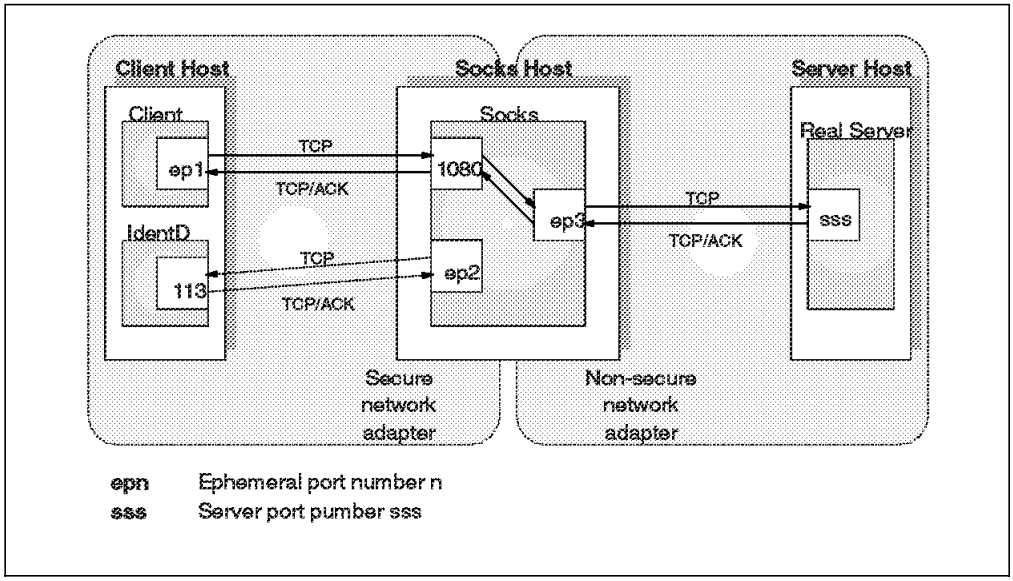


Figure 134. Socks TCP Segment Flow

When you enable a socks server on your gateway between a secure and a non-secure network, you need to add IP filter rules that will allow the use of the socks server.

In general, you need to add three sets of rules:

1. Allow hosts in the secure network to connect to the socks server on the OS/390 host.
2. If the Ident protocol is used, then allow the OS/390 host to connect to the IdentD server on hosts in the secure network.
3. Allow the socks server on the OS/390 host to connect to any TCP server port number in the non-secure network.

5.9 Domain Name Resolution (DNS)

The Domain Name System (DNS) application can be a challenge when you want to set up a connection between a secure and a non-secure network. DNS is primarily used to resolve host names into IP addresses and IP addresses into host names. DNS is also used for other purposes, such as finding out where to deliver mail for a specified destination host.

The DNS protocol uses both UDP and TCP. When a host wants to query a name server, it uses UDP. When two name servers that act as primary and secondary name servers for the same name space (called a zone in DNS terminology) need to exchange information with each other, the secondary name server connects to the primary name server and uses TCP for bulk transfer of the zone data.

You would normally not do zone transfers in or out of your secure network, so we really only need to focus on DNS′s use of UDP for name queries.

Another aspect of using DNS is that you want to allow your secure hosts to resolve both secure and non-secure host names, but you do not want non-secure hosts to be able to resolve your secure host names.

OS/390 Firewall Technologies does not use a special name server on OS/390, but uses the standard OS/390 UNIX System Services name server that comes with Communications Server for OS/390. The configuration of the name server is done through OS/390 Firewall Technologies configuration commands.

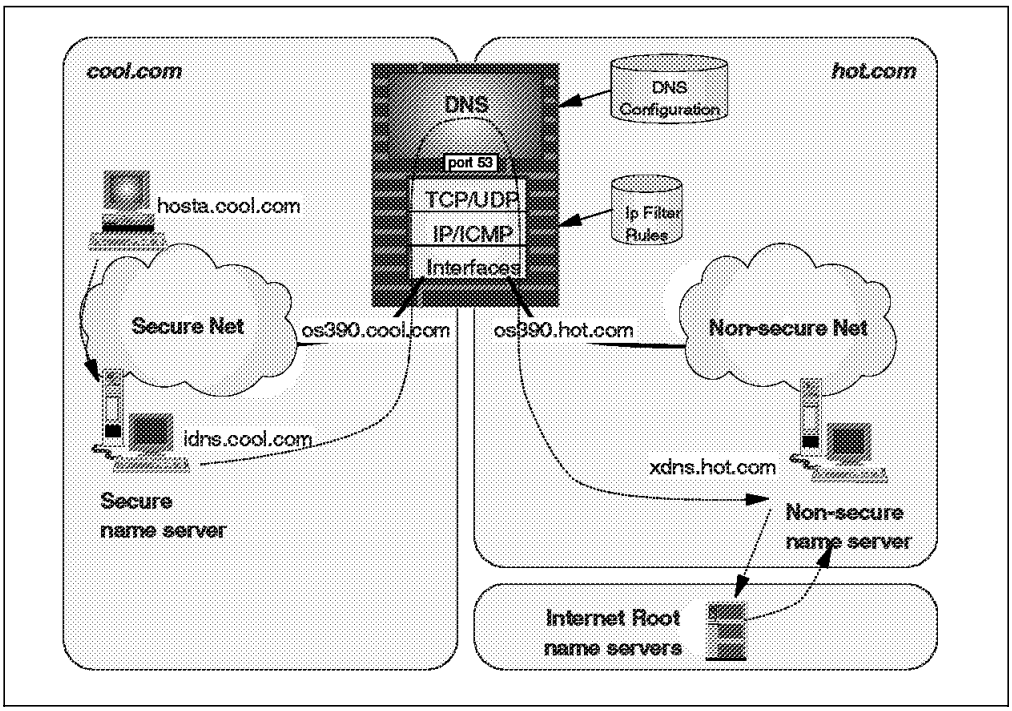


Figure 135. DNS Setup Overview

The job of the name server on the gateway between the secure and the non-secure network is to forward queries from the secure network to name servers in the non-secure network, and to cache and return the responses from the non-secure network to the name server in the secure network.

If an application program runs on the gateway itself, it should generally be considered part of the secure network, and a name query should be directed to the internal name server, which then forwards the query through the DNS on the gateway to the non-secure network. This is accomplished through proper configuration of the local resolver configuration file (resolv.conf) on the gateway.

Seen from the secure network, the name space in Figure 135 consists of the cool.com domain as well as all other domains defined in the Internet.

Seen from the Internet, only the hot.com domain exists. The hot.com domain in this example only has two entries: one for the external name server host itself (xdns.hot.com), and one for the gateway (os390.hot.com).

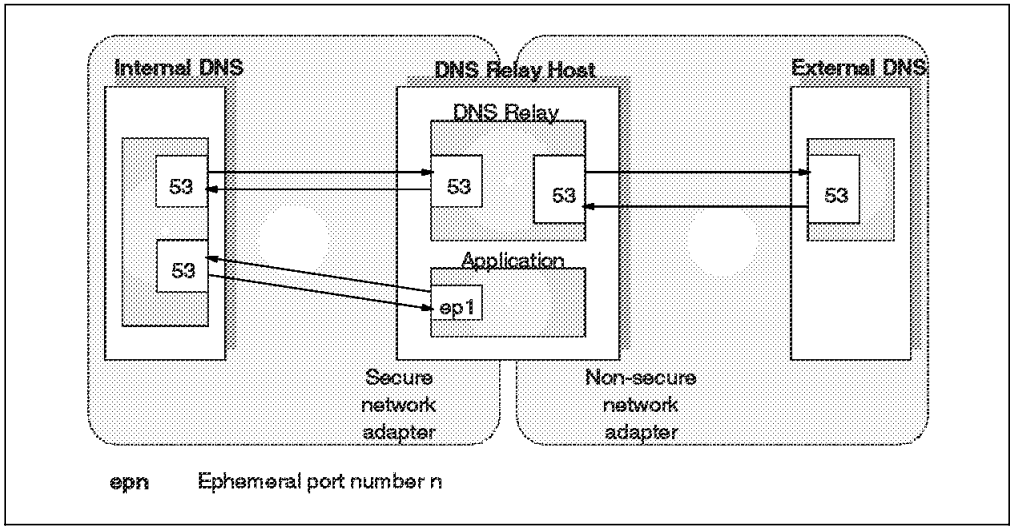


Figure 136. DNS UDP Datagram Flow

When a name server queries another name server, it sends UDP datagrams from port 53 and to port 53. In a configuration, such as in Figure 135 on page 145, where the name server on the gateway between the secure and the non-secure network only acts as a relay, you can work with rather simple IP filter rules that basically allow UDP datagrams to and from port 53 to flow over both the secure and the non-secure adapter. Over the secure adapter you also need a rule that allows local applications on the gateway itself to query the internal name server. Such queries from a program (or, more precisely, the resolver code that is linked with a program) come from an ephemeral port number to the name server port number 53.

The configuration in Figure 135 on page 145 is called a *merge name server* configuration. The name server on the gateway does not have authority over any zone data, but only acts as a relay (a forwarder) between the secure name server and the non-secure name servers.

You are able to work with more complex configurations in which the name server on the gateway, in addition to the preceding role, also acts as the external name server itself. If you wish to avoid having another machine in the non-secure network just to host your external name server, you can configure the name server on the OS/390 gateway as the external name server. You would then have to add an extra filter rule that would allow an ephemeral port number in the non-secure network to send queries to port 53 on the gateway over the non-secure adapter.

Chapter 6. Network Computing Enhancements

This chapter describes the enhancements made to Network Computing with Release 5 of OS/390. The enhancements to the Network Computing components are as follows:

- Domino Go Webserver 4.6.1
  - %%CERTIF%% support

    %%CERTIF%% adds new access control for Domino Go Webserver, allowing an administrator to define an MVS user ID and its associated access capabilities for a particular request. %%CERTIF%% allows a browser to present a certificate to the Domino Go Webserver, which is then used to establish the associated MVS user ID to process the request.

    With this support, Unix System Services can receive a certificate from the Domino Go Webserver and copy it to system storage. The server can then use the certificate in place of a user ID and password. UNIX System Services recognizes that a certificate has been passed and invokes RACF′s InitACEE callable service, passing the certificate itself (fullword length plus certificate) and a new flag that says “certificate being passed.” The InitACEE service will decode the certificate, do the profile lookup, and extract the user ID. Then InitACEE functions it does currently: it will check the cache for a previously created ACEE and, if not found, do a RACINIT.
  - Java servlet support

    Java servlets (internal and external processes) are now supported under MVS, based on the latest available JDK 1.1. (Java CGI support was enabled with ICSS V2.2 using JDK 1.02.)
  - Performance enhancements

    General performance improvements in the base server and in the processing of secure transactions have been made.
- Communications Server

  The following enhancements are made to TCP/IP:
  - New IP communications stack

    This offers substantially higher levels of performance and reliability for the most popular TCP/IP applications and APIs, and replaces TCP/IP 3.2 in OS/390. Communications Server (CS) for OS/390 Release 4 included an entirely new communications stack that provided substantially improved performance, reliability, availability, and serviceability for OS/390 users running UNIX applications with TCP/IP networks. These benefits are extended to all other popular TCP/IP Application Programming Interfaces (APIs) and applications in Release 5. The TCP/IP Version 3 Release 2 function previously included in CS OS/390 is supplanted by this new support and is no longer included in Communications Server for OS/390.
  - New Telnet server provides TN3270E

    A new Telnet Server is developed that uses UNIX Services sockets to communicate with the TCP/IP stack. The new Telnet Server provides the

same function that is available with the non-UNIX Telnet server in TCP/IP V3.2, plus the following new functions:

- RFC1647 (TN3270E) support
- Dynamic updating of Telnet configuration statements
- New Telnet Server operator commands for start, stop, configuration changes and display
- Improved Telnet Server performance and user connection load
- Improved MVS RAS characteristics

– Domain Name Server Support

This is integrated into Release 5 of OS/390. (For Release 4, it was separately orderable).

– Workload balancing

TCP/IP users can now take advantage of workload balancing and XCF communications of S/390 Parallel Sysplex Servers.

– Multiprotocol performance is improved.

– Native ATM support is now available also for TCP/IP users.

– Additional enhancements are included for SNA/APPN and HPR users.

• Integrated Cryptographic Support Facility (ICSF).

OS/390 Version 2 Release 5 ICSF includes support for the following:

– Triple DES encipherment for data privacy

Triple DES uses a triple-length data-encrypting key to encipher and decipher data. Triple DES encipherment is superior to single encipherment because it increases the work needed to break the cipher and provides extra protection for enciphered data.

– The Secure Electronic Transaction (SET) protocol

The SET protocol was developed jointly by Visa International and MasterCard for safeguarding bank card purchases made over open networks.

– A public key data set (PKDS)

The PKDS is a VSAM data set in which an application can store both RSA and DSS public and private keys. ICSF provides PKDS update callable services for creating and writing records to the PKDS and reading and deleting records from the PKDS. Callable services that accept public key tokens as input can now also accept key labels.

– Zero-pad for DES key exchange with RSA

Zero-pad is an alternative to the Public-Key Cryptography Standard (PKSC 1.2) padding rules. For installations that currently use padding with zeros for DES key exchange with RSA, zero-pad provides a path to migrate from the 4753 IBM Network Security Processor to the S/390 Enterprise Servers.

– Visa Card Verification Value and MasterCard Card Verification Code generation and verification

The Visa CVV and the MasterCard CVC are cryptographically-generated values that can be used to detect forged credit cards.

6.1 The Value of OS/390 as a Web Server

IBM made its first World Wide Web (WWW) server for MVS available in December 1995. That product was the IBM Internet Connection Server for MVS/ESA.

Many people are still surprised that MVS (or OS/390, as the environment is now known) can be a Web server. Some people would have you believe that Internet servers must be UNIX or similar systems. However, you will find significant value in putting your Web server on OS/390. OS/390 already holds significant amounts of your business data, which could be made more available to the people who need it using Internet technologies. Further, since OS/390 is a critical business system for so many enterprises, it has built-in strengths such as very high capacity, availability, security, and integrity. These strengths can be applied to Internet work just as they are applied today to transaction processing, very large databases, and batch processing.

By now, we have seen four releases of the Web server, each one offering more function and more System/390 exploitation than the previous release.

OS/390 Release 5 now integrates the Lotus Domino Go Webserver Version 4.6.1. We begin with a brief review of Web technology and its value for OS/390 users.

6.1.1 The Value of the World Wide Web

There is no shortage of information available on the value of the Web. It is not our intention to replicate that information here, but we do feel it is worthwhile to provide a short summary.

The Web provides a light-weight client/server architecture. Following the introduction of networked personal computers (with Graphical User Interfaces) to the workplace, the client/server model of computing was defined and popularized. The idea behind client/server is that some proportion of a computing application is performed on a client machine such as a PC, and some proportion on a server (like MVS). This allows use of the localized power and speed of the client machine for the parts of an application where it makes sense, and for presentation of application information using a Graphical User Interface (GUI). However, time has shown that the client/server computing model is an expensive one and applications are often difficult to implement. Software distribution and management are key components of the costs.

The architecture behind the Web was introduced in 1991. It was designed to allow simple document distribution and cross-referencing for researchers working on the Internet. It is based on a lightweight request and response protocol, where a client (Web browser) makes a request to a Web server. Generally, requests are made for files that contain formatting commands and text, or multimedia content such as graphics, sound, or movies. These files are displayed on the Web browser, often in a graphical and attractive way.

The widespread availability of graphical Web browsers, TCP/IP stacks for personal computers, and cheap and easy access to the Internet (in most parts of the world) have made the Web the most rapidly adopted technology in recent years. It is simple to connect to the Internet, simple to use, and many enterprises are now providing services using the Web.

The value of the Web, therefore, is that:

- It is simple and cheap for *clients* to connect to and use the vast array of information and applications available. Therefore there are many clients. Organizations such as Matrix Information and Directory Services (`http://www.mids.org`) have estimated a consumer client population of 57 million in January 1997, with the number growing rapidly.
- Clients can access information 24 hours per day.
- Enterprises widen their potential market to a worldwide audience.

  Because of reasonably strong standards, enterprises can provide information and applications on the Web, being confident that the vast majority of clients will be able to use that information.
- The provider of information and applications on a server is freed from the responsibility of supplying and managing client software for their application or information. As a direct consequence of this, the enterprise is also freed of responsibility for installing and maintaining a client hardware platform and operating system, performing capacity planning and tuning, problem analysis, and so on, for the client systems. In other words, the costs of deploying the client side of a client/server application using this technology are lower than using other techniques.

6.1.2 The Value of OS/390

Most OS/390 (and MVS) users are already familiar with the strengths of the OS/390 environment. It is no accident that many of the world′s largest banks, insurance companies, government agencies, and other large enterprises use MVS or OS/390. A detailed treatment of this subject is available in *Selecting a Server - The Value of S/390*, SG24-4812. Some of the key characteristics of OS/390 that make it a valuable server for Web applications are:

Scalability: OS/390 (and before it, MVS) has an underlying design which allows a server to scale up to process very large amounts of work. Since the introduction of System/360 in 1964, constant design enhancements have added scalability to the platform. Significant enhancements have included virtual storage addressing, 31 bit addressing, access register addressing (data spaces), efficient use of symmetric multiprocessors, and the Parallel Sysplex architecture.

The use of the Workload Manager was introduced in OS/390 Release 4. This is an example of exploitation of the scalable characteristics of OS/390. Although DGW is not currently a sysplex data sharing application, the existence of the underlying sysplex architecture allows exploitation of sysplex when required. More information on Parallel Sysplex is available in *A Comparison of System/390 Configurations - Parallel and Traditional*, SG24-4514.

Availability: OS/390 systems are used in environments where high availability is crucial. For example, many banks run their ATM machines from OS/390 servers, with very high levels of availability. For Web applications, availability becomes very important for several reasons, including:

- The Web “culture” has caused us to become very impatient consumers of information. If a particular Web site is not available or is responding slowly, clients will tend to shift their attention to another site, perhaps that of a competitor.
- The Web extends the “hours of opening” for your enterprise to 24 hours a day, since clients can be spread across the world. In addition, even in your

own time zone, clients will require connections to your site outside your regular business hours.

OS/390 servers are already designed and configured to deliver the levels of system availability required, especially in a Parallel Sysplex environment.

Security and Integrity: A major concern of commercial and government enterprises is the security and integrity of their data. This concern is magnified when connections to the Internet, or even extended access through an intranet, are considered. OS/390 provides a very strong framework for control of data access, and for maintaining the integrity of data. IBM announced an integrity guarantee for MVS in 1972, and the design of System/390 systems is such that data integrity is carefully maintained. Security is controlled by the System Authorization Facility (SAF), which checks all access requests to system resources, using an external security manager (such as the OS/390 Security Server, formerly known as RACF) to grant or deny access to resources.

All requests made to DGW are given a user ID, and work is done in the server on behalf of that user ID. Therefore you can use the Security Server to restrict access to system resources, and be confident that a Web user cannot compromise the security of your system.

IBM has conducted penetration tests on DGW and found it significantly more difficult to break in to than other systems. IBM has expertise in running penetration tests. If you are interested in running a penetration test on your own server, contact your IBM representative who will contact IBM Consulting for you.

Operational Applications and Data: One of the greatest strengths of System/390 servers is the amount of operational data, and the number of operational applications, that exist today on those servers. Many of the applications have been running in production for many years, and they deliver significant business value for large enterprises. For example, most large banks run transaction systems, which perform most of the bank's business, on System/390 servers.

As the *OS/390 Internet BonusPak II* showed, it is possible to connect those applications and that data to Web clients using DGW on OS/390. Product gateways exist to allow you to access DB2 data, CICS transactions, MQSeries applications, and BookManager books. Sample programs are available which demonstrate access to other resources, such as IMS Transaction Manager applications.

Support for Open Standards: The OS/390 system of 1997 is very different from the MVS of 1990. While retaining the strengths we have already discussed, OS/390 has been branded to the UNIX'95 level by the X/Open Company Limited (see `http://www.s390.ibm.com/stories/unix95.html`). Many UNIX applications have been successfully ported to OS/390. In addition, capability exists to port Windows NT applications. Open networking protocols such as TCP/IP are available, and work is in progress to bring the Lotus Domino server to OS/390. The system is built to run multiple different types of work, all at the same time, with workload balancing and management performed by the system.

Conformance to standards simplifies the process of porting, and therefore increases the speed at which applications can be ported. Support for standards also allows ported applications to use some "traditional" System/390 attributes transparently and automatically. Examples of automatic exploitation include the use of the OS/390 address space architecture. System/390 extensions (within

the standardized framework) allow the evolution of “UNIX style” applications to take advantage of System/390 attributes through specific exploitation. For example, UNIX applications can take advantage of System/390 security, workload management, operations, and measurement and tuning interfaces by including extensions that use those services.

6.1.3 The Strengths of DGW 4.6.1 as a Web Server

Lotus Domino Go Webserver 4.6.1 is an equivalent functional level of Web server to the IBM Internet Connection Secure Servers at level 4.2, for OS/2, Windows, AIX, and so on. The IBM servers provide very strong functionality, allowing you to build modern and robust Web sites. More information about this level is available from `http://www.ics.raleigh.ibm.com/`.

This section describes some of the key functions of DGW that can deliver significant value.

Secure Sockets Layer (SSL) Version 3: SSL is an industry standard protocol for transmitting secure information over an insecure network, such as the Internet. Essentially, SSL provides a means for encrypting data that is transmitted. Significantly, however, SSL also provides a negotiation protocol so that two parties who have not previously agreed on encryption schemes or keys, can do so at run time.

SSL Version 3 provides enhancements over the previously supported version, including client certification and improvements to the protocol to minimize the likelihood of unauthorized access to the data you are sending.

Client certification is an important prerequisite for electronic commerce. When both the client and the server have provable identities (contained in certificates issued by a trusted certification authority), transactions can be conducted.

Secure Sockets Layer Tunneling: Tunneling is a technique which allows a Web client to connect to a secure server (using SSL) through a proxy server. If the proxy supports SSL tunneling, it passes on the request without examining it or requiring decryption and encryption.

Application Programming Interfaces: In addition to the generic Common Gateway Interface (CGI), DGW provides a modern, high performance API, the Internet Connection API (ICAPI). ICAPI was introduced with Version 2 Release 1. of ICSS, and is enhanced with new services, extra diagnostic capabilities and messages, and the inclusion of ASCII-to-EBCDIC conversion.

An ICAPI module is also supplied to provide support for Netscape API (NSAPI) programs. Many NSAPI programs can be re-compiled on OS/390 and used as is, without the need to rewrite them. This allows you to more easily port existing Netscape Web programs to OS/390.

In summary, you can use server programs in multiple languages and architectures. You can extend your server functionality in Java, C, C++, COBOL, PL/I, Rexx, NetRexx, Perl, Shell Script, and other languages. You can write or port CGI programs, ICAPI programs, or recompile NSAPI programs to run on the server.

HyperText Transfer Protocol (HTTP) 1.1: HTTP is the protocol that manages the communication between clients and servers on the Web. The “version” in common use today is 1.0, but it does not have formal status as an Internet protocol.

The HTTP protocol is promoted by the World Wide Web Consortium (W3C). The W3C has been focusing recently on improving the HTTP protocol so that Web work is less dominant on the Internet, and the protocol is accepted as an Internet standard. To these ends, the 1.1 level of HTTP has been defined and submitted to the Internet Engineering Task Force (IETF). It includes enhancements to improve performance, and to enable one IP address to support multiple servers. This level is supported by DGW, and the level of HTTP used is negotiated between the client and the server.

Platform for Independent Content Selection (PICS): An emerging issue with the Internet is the widespread availability of material that is considered inappropriate or offensive to many people. There is so much concern that some National Governments have enacted legislation, or otherwise imposed restrictions, that limit material being published on the Internet. Such legislation, however, is limited in effectiveness, while at the same time imposing restrictions which some consider to be in conflict with their rights (for example, the right to free speech in the United States).

PICS is a technology solution which allows the client to decide which material they consider appropriate. It is a labelling scheme that allows Web sites to rate their own content, also making it easy for interested individuals or organizations to provide an “independent” rating service. The W3C expects rating services to be built using this technology, and that Independent Service Providers will provide personalized “blockades” to offensive content based on your preferences, using proxy servers enabled with PICS filters. For businesses, it will become very important to understand PICS, and ensure your site is rated correctly, to maximize your audience. If your business provides content that may be considered offensive by some people, providing appropriate ratings *may* also reduce your legal exposure.

Workload Management: DGW exploits the OS/390 Workload Manager function. Web work can be classified as it enters the system, and can be given performance goals in business terms. This means that you can determine the importance of various types of Web work to your business, relative to all other types of work in your system, and set priorities accordingly for allocation of scarce system resources, which will be managed automatically. This also means that your Web server, and the transaction systems and business data it accesses, can be on one system, which is more efficient and less subject to arbitrary network delays. Most other systems require manual tuning, and very few other system vendors recommend running multiple different types of work on one system.

This ability minimizes your system management costs, and allows you to use all of the capacity of your server. For example, when transaction systems are running during business hours, Web work can be set to a different service level so that it does not interfere with the more important business transactions.

This facility also gives you the flexibility to configure multiple Web server address spaces, each to process different types of work. Server address spaces are automatically started when required by the Workload Manager, and can be

automatically restarted after a failure. Web work can be classified to allocate different performance characteristics to different types of work.

In other words, not only can you distinguish Web work from transaction processing work, you can also distinguish simple HTTP GET requests from ICAPI database lookup requests, for example.

Access to MVS Data Sets: In addition to providing access to data stored in the Hierarchical File System (HFS) on OS/390, DGW can now retrieve data from sequential and partitioned data sets. This means that OS/390 customers can store Web content in files they are familiar with.

Logging and Statistical Reporting: Many enterprises are interested in understanding who visits their Web site, and what they do during a visit. If your interest is in advertising your services, you will also be interested in how effective your message is. Many Web site designers use logging information to adjust the design of the site to make it easier to use and navigate. DGW provides both industry-standard and extended logging and reporting tools to help you determine how effective your site is, and who is using it.

The extended reporting tools provided by DGW include a Java applet-based tool that allows more detailed analysis of logs than previous tools, and a Web Usage Mining tool, which allows you to analyze the behavior of users that visit your site.

OS/390 Console Support: DGW can be controlled from the system operator's console. Stop and modify commands are available. In addition, OS/390 DGW messages have been enhanced to provide information to automation routines, should you require an automation package to manage your Web servers.

6.1.4 The Strengths of OS/390 As a Web Server

These strengths include:

- The ability to set performance goals for your Web work, along with performance goals for other work in the system, and have the system allocate resources according to those performance goals. This function is provided by workload management.
- Strong system security, which lowers the likelihood of successful hacking attacks on your system. A well administered OS/390 system is very difficult to break into.

6.1.5 OS/390 and the Web - Delivering Applications

It is clear from the preceding discussion that OS/390 makes an excellent Web server, joining the strengths of OS/390 with the strengths of the modern and feature-rich DGW product. The combination of OS/390 system infrastructure and DGW gives advantages that are not available in other product sets.

The introduction of DGW to your OS/390 environment is the first step in enabling you to connect your clients directly to the applications and data that are of interest to them. Some examples of applications that have been proposed include:

- Allowing citizens to submit their self-assessment tax returns directly from a Web browser to the taxation authority's system, minimizing paper processing and manual data entry.

- Allowing any Web client to look up a telephone number from a telephone company′s database.
- Allowing customers to query their bank balances from their Web browser.
- Providing browse and buy capabilities to a mail-order company.
- Providing travel agency services, such as transportation and accommodation searches, reservations, and tickets.

Undoubtedly, you will be able to provide your own examples of applications that, if made available directly to clients or partners, would enable you to reach new markets, provide new services, or reduce your costs.

The advantages of doing this on OS/390 are described in 6.1.2, “The Value of OS/390” on page 150. If your applications and data are already on OS/390, directly connecting to them is simpler and cheaper than installing an additional server to enable the connection. If you expect your Web application will attract many customers, System/390 can provide the large scale, high availability and security that you require.

These factors make OS/390 with DGW an excellent choice for your Internet and intranet server. The more mission-critical your Web site will be, the more you require a mission-critical server platform.

The early stages of your use of the Internet are characterized by activities such as deployment of inter-enterprise mail or the establishment of a simple Web site. These activities may be considered important, but perhaps are not considered mission-critical in the same way as a transaction system. As your Internet presence (or your intranet services) become more important in terms of business impact, so the importance of your server increases. As you reach the point where your network applications and data become mission-critical, so the value of OS/390 becomes apparent as a server capable of delivering high availability and consistently high performance.

For Web applications that are crucial to your enterprise, OS/390 provides the performance, availability, security, and integrity that you need.

6.2 Domino Go Webserver Version 4 for OS/390

Domino Go Webserver 4.6.1 for OS/390 is a replacement product for the Internet Connection Secure Server. DGW adds to functions already available in Domino Go Webserver 4.6 for OS/390. DGW operates on all ESA-capable machines supporting OS/390 release 4 and later. As DGW is a replacement of ICSS 2.2, it supports all the features in ICSS 2.2 and also has the following enhancements:

- Authentication of a requestor using a certificate instead of a user-entered ID and password
- Running Java servlets in a S/390 host using Java 1.1 support
- Performance enhancements which result in a throughput increase
- Proxy enhancements which result in improved server RAS
- Fast Common Gateway Interface (CGI) which reduces latency

The performance and proxy enhancements to Domino Go Webserver 4.6.1 for OS/390 are transparent to the MVS systems programmer and as such, no installation or implementation work has to be done for these enhancements.

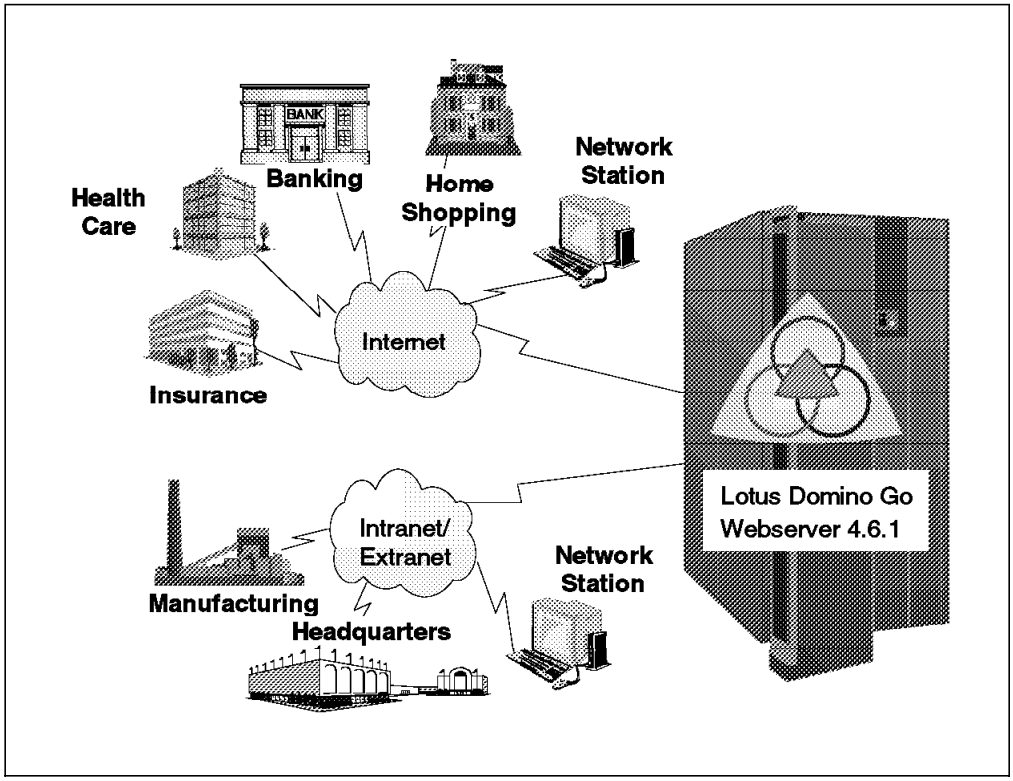


Figure 137. OS/390 Domino Go WebServer 4.6.1

6.2.1 Installing and Implementing Domino Go Webserver 4.6.1 for OS/390

The installation of Domino Go Webserver 4.6.1 for OS/390 is very well covered in the publications that are supplied with the base product:

- *Domino Go Webserver 4.6.1 Planning for Installation*
- *Domino Go Webserver 4.6.1 Webmaster's Guide*

If this is the first time that the installation of a Web Server for OS/390 is being implemented, it is strongly recommended that you read the Redbook *Enterprise Web Serving with Lotus Domino Go Webserver for OS/390*, SG24-2074 that will be available in September of 1998.

6.2.2 Implementing Authentication Using a Certificate

The enhancement to Domino Go Webserver 4.6.1 for OS/390 allows Domino Go Webserver to authenticate to OS/390 an SSL requestor using a certificate, issued by a recognized authority, instead of a user typing in their user ID and password. OS/390 Security Server (RACF) Release 5 incorporates the digital certificate support.

In a client/server environment network, clients identify themselves using digital certificates. The Domino Go Webserver (DGWS) authenticates a client using the client's certificate and the Secure Sockets Layer (SSL) protocol. This support enables the creation of accessor environment elements (ACEEs) for RACF-defined user IDs based on information contained within the client certificates. This means that the RACF user ID and password of each client do not need to be supplied when accessing secure Web pages.

A digital certificate or digital ID, issued by a certifying authority, contains information that uniquely identifies the client. This identifying information consists of the certifying authority′s name and serial number and the subject′s name and public key. After a user has supplied the digital certificate to DGW, DGW checks that the certificate is valid. If the certificate is valid, the user does not need to be re-authenticated by entering a RACF user ID and password.

To make this digital certification possible, DGW, UNIX System Services, and RACF perform the following tasks:

- DGW passes the client′s digital certificate to UNIX System Services. If DGW has at least READ authority to the BPX.SERVER resource in the FACILITY class, it is assumed that the SSL protocol was used. This means that the server has verified that the certificate is genuine and was issued by a trusted authority, that its validity dates are current, and that the client is the owner of the certificate.
- UNIX System Services passes the information to RACF.
- RACF extracts information from the digital certificate, identifies a RACF user ID from it, and builds an ACEE.

6.2.3 Java Servlets

The following list describes the attributes of servlets.

- They are ordinary Java programs that use additional packages found in the Java Servlet API.
- They run on a Web server machine inside a Java-enabled server.
- They extend the capabilities of the Web server.
- They can be loaded automatically when the Web server is started.
- They can be loaded when the first client requests the service of the servlet.
- They can stay running waiting for additional client requests.
- They extend the capabilities of the server by creating a framework for providing request/response services over the Web.

The following is an example of how the servlets could be used for providing request/response services over the Web.

A client sends a request to the server. The server sends the request information to the servlet. The servlet then constructs a response that the server sends back to the client.

The servlet can use all the capabilities of the Java language in constructing the response as it is a Java program.

The servlet can also interact with outside resources, such as files or databases or other applications (also written in Java, or other languages), to construct the response and possibly to save information about the request/response interaction.

The response to the client, therefore, can be a dynamic and unique response to the particular interaction, rather than an existing static HTML page.

The packages supplied when installing JDK with servlet support are the following:

- javax.servlet

- javax.servlet.http

Within the two packages are seven interfaces, five classes, and two exceptions. These interfaces, classes, and exceptions powerfully extend the capability of Java.

6.2.3.1 Java Servlets Implementation

When implementing servlets we assumed that DGW had already been installed. If DGW has not been installed follow the installation documentation supplied in *Domino Go Webserver 4.6.1 Planning for Installation*. Once DGW had been installed, we did the following to enable and implement servlet support:

- Install the latest JDK available from the Web site mentioned in *Domino Go Webserver 4.6.1 Planning for Installation*.
  - Download the JDK tar file from the Web site.
  - Install the JDK tar file as documented in the installation instructions supplied on the Web site.
- Change the following files in UNIX System Services:
  - /etc/profile
  - /etc/http.conf
  - /etc/http.envvars
  - /etc/servlet.conf
- Compile the sample servlets.
- Stop and start the Web server.
- Invoke a sample servlet.

6.2.3.2 Impact on Profile file

Shown in Figure 138 is a sample profile file from the system where we implemented servlets.

```
 .
 .
 .
echo ------------------------------------------------------------
echo Set up Environment Variables for Java and Servlets for OS/390 -
echo ------------------------------------------------------------
unset CLASSPATH
#
PATH=/usr/lpp/Java/J1.1/bin:$PATH
export PATH
echo  PATH reset to $PATH
#
CLASSPATH=/usr/lpp/Java/J1.1/lib/classes.zip:
/usr/lpp/internet/server_root/cgi-bin/icsclass.zip:
/usr/lpp/internet/server_root/servlets/public
export CLASSPATH
echo  CLASSPATH reset to $CLASSPATH
echo ------------------------------------------------
```

Figure 138. /etc/profile Sample

As documented in the JDK installation documents, we had to change the PATH environment variable and then we had to set the CLASSPATH environment variable. The CLASSPATH statement shown in Figure 138 is not a true depiction, as it is contained on one line in the /etc/profile file in UNIX System Services. We have shown it this way for clarity. We set the CLASSPATH environment variable to contain the following three directories:

- The directory that contains the Java classes:

  /usr/lpp/Java/J1.1/lib/classes.zip

- The directory that contains the servlet API classes:

  /usr/lpp/internet/server_root/cgi-bin/icsclass.zip

- The directory where the servlet classes reside:

  /usr/lpp/internet/server_root/servlets/public

Java was installed in the following directory:

- /usr/lpp/Java/J1.1

This directory was the JAVA_HOME directory for our installation.

6.2.3.3 Impact on http.conf File

In Figure 139, the Java servlet support directives are shown. The Java servlet support directives are contained in the http.conf file. We made no changes to the supplied directives in http.conf. If a different servlet.conf file is to be used, the ServerInit directive in http.conf can be changed. If the ServerInit directive is changed, remember to change the EServlet directive in servlet.conf as well. The Java servlet support directives are documented in the *Webmaster's Guide*.

```
 .
 .
 .
#   ======================================
#   *** JAVA Servlet Support directives ***
#   ======================================
ServerInit      /usr/lpp/internet/bin/libjavsetup.so
                :Setup /etc/servlet.conf
ServerInit      /usr/lpp/internet/sbin/libservlet.so:Init WEBSRV
#ServerInit      /usr/lpp/internet/bin/libintserv.so:ServletInit
#ServerTerm      /usr/lpp/internet/bin/libintserv.so:ServletTerm
ServerTerm      /usr/lpp/internet/sbin/libservlet.so:Term

#Service         /servlet/*
/usr/lpp/internet/bin/libintserv.so:ServletService*
Service         /servlet/*
                   /usr/lpp/internet/sbin/libservlet.so:servlet*
Service         /admin-bin/RestartJVM
                   /usr/lpp/internet/sbin/libservlet.so:RestartJVM
```

Figure 139. /etc/http.conf Sample

6.2.3.4 Impact on http.envvars File

The sample http.envvars, as shown in Figure 140, was not changed for implementing servlets. When installing DGW, the statements in http.envvars are set when setup.sh is executed.

```
PATH=/bin:.:/usr/sbin:/usr/lpp/internet/bin
           :/usr/lpp/internet/sbin
           :/usr/lpp/Java/J1.1/bin
SHELL=/bin/sh
TZ=EST5EDT
LANG=C
LC_ALL=en_US.IBM-1047
NLSPATH=/usr/lib/nls/msg/%L/%N:/usr/lpp/internet/%L/%N
LIBPATH=/usr/lpp/internet/bin
        :/usr/lpp/internet/sbin
        :/usr/lpp/Java/J1.1/lib/mvs/native_threads
JAVA_HOME=/usr/lpp/Java/J1.1
CLASSPATH=/usr/lpp/Java/J1.1/lib/classes.zip
         :/usr/lpp/internet/server_root/cgi-bin/icsclass.zip
         :/usr/lpp/internet/server_root/servlets/public
         :/usr/lpp/internet/server_root/cgi-bin
STEPLIB=CURRENT
```

Figure 140. /etc/http.envvars Sample

6.2.3.5 Impact on servlet.conf File

In Figure 141, a sample servlet.conf file is shown. The JavaClassPath directive is contained on one line in the file but is shown here across multiple lines for clarity. In our implementation of servlet support, the JAVA_HOME directory was /usr/lpp/Java/J1.1.

```
 .
 .
 .
MaxActiveJavaThreads  10
IcsjProcessKillTimeout  5
ServletLog     /usr/lpp/internet/server_root/logs/servlet-log
JVMLog     /usr/lpp/internet/server_root/logs/jvm-log
JavaPath           /usr/lpp/Java/J1.1/bin
JavaLibPath        /usr/lpp/Java/J1.1/lib/mvs/native_threads
JavaClassPath      /usr/lpp/Java/J1.1/lib/classes.zip
              :/usr/lpp/internet/server_root/cgi-bin/icsclass.zip
              :/usr/lpp/internet/server_root/servlets/public
EServlet ReqInfoServlet ReqInfoServlet {
}
EServlet extConfigServlet COM.lotus.go.admin.ExtSvltCfgServlet {
         svltconfig=/etc/servlet.conf
}
```

Figure 141. /etc/servlet.conf Sample

Using the Java servlet configuration directives, you have the ability to:

- Turn the servlet support on or off

- Specify the number of Java threads to use to process servlet request
- Name the directory where you keep the servlets
- Choose whether to log servlet messages and the location of the log
- Specify a servlet's initialization parameters

The Javapath directive specifies in which directory the Java Developers Kit executable file is located. The value for this directive is JAVA_HOME/bin. In the sample shown in Figure 141 on page 160, this translates to /usr/lpp/Java/J1.1/bin.

The JavaClassPath directive specifies in which directory the Java class files are located. In our implementation we had three class files specified and they are shown in Figure 141 on page 160.

The MaxActiveJavaThreads directive is used to specify the maximum number of Java servlet request threads that will run in the external Java Virtual machine (JVM) process.

The ServletLog directive is used to specify the name and location of the log file for external and internal Java servlet messages.

The ESerlet directive specifies the instance name, the class name of the servlet and the value of the parameters passed to a Java servlet when the servlet is initialized. In Figure 141 on page 160 we have specified two EServlet directives. The EServlet directive for extConfigServlet is the servlet used when accessing the WebServer External Servlet Configuration from the Web browser.

6.2.3.6 Compiling a Servlet

The Domino Go Webserver is installed in a specific location. In our installation, this directory was /usr/lpp/internet/server_root. This is commonly refered to as the server_root directory. To be able to use the sample servlet Java files the servlet class files must reside in server_root/servlets/public. The sample servlet Java files, are supplied in server_root/servlets/public. Four samples are provided as documented in the *Web Programming Guide*.

- ReqInfoServlet
- FileToBrowserServlet
- FormDisplayServlet
- FormProcessingServlet

To be able to compile the servlets, go to the directory where the sample servlets are installed and issue the following command:

- JAVAC servlet name (for example, JAVAC ReqInfoServlet)

A Java class file with the file name of ReqInfoServlet.class is then produced and this is the servlet.

6.2.3.7 Stopping and Starting the Webserver

The DGW only has to be stopped and restarted to access the sample servlets if the servlets have been used before. To stop the DGW, issue the MVS stop command for the Webserver. Once the Webserver is stopped, issue the start command to start the Webserver.

To restart the Webserver, issue the OS/390 modify command as shown in Figure 142. The various console commands for the DGW are documented in the *Webmaster's Guide*.

```
F IMWEBRHR,APPL=-restart
```

Figure 142. Sample Webserver Restart Command

Once the configuration changes have been made for servlet support and the DGW has been stopped and started, the trace entries shown in Figure 143 can be seen. These trace entries indicate that the servlet support is implemented correctly.

```
  .
  .
  .
A166FF8 at 07/Apr/1998:12:51:12 +0500
 : Init: begin external servlet initialization
  .
  .
  .
A166FF8 at 07/Apr/1998:12:51:44 +0500
 Init: servlet initialization complete: rc=200
  .
  .
  .
A456618 at 07/Apr/1998:12:51:54 +0500
 Ext servlet trace..  External servlet initialization was successful.
  .
  .
  .
```

Figure 143. Sample Trace Entries Indicating Servlet Initialization

6.2.3.8 Accessing the Servlet

The use of the servlet directory is similar to the use of the document root directory. The server expects to find servlet class files in the servlet root directory, in server_root/servlet/public. However, you must specify a symbolic servlet to find the servlet. For example, to access the servlet directly from a browser by specifying the servlet URL, you enter:

```
http://your.server.name/servlet/ReqInfoServlet
```

A default symbolic servlet that stands for the location server_root/servlets/public must be specified. Do not specify the class extension for ReqInfoServlet. The servlet name is case-sensitive. Also note that all servlets must be in the servlet directory, which is the server_root/servlets/public location. You cannot place servlets in a subdirectory of the servlet directory and access them there, unless

you add the subdirectory to the JavaClassPath directive in the DGW configuration file.

The implementation and installation of the sample servlets are complete once the sample servlets are accessed from a Web browser using the DGWS.

Chapter 7. SMP/E Enhancements

SMP/E R5 has various new enhancements implemented in OS/390 R5. In the following chapter we only discuss the following new enhancements:

- SMPPTS data set compaction
- Enhanced receive processing
- Library change interface

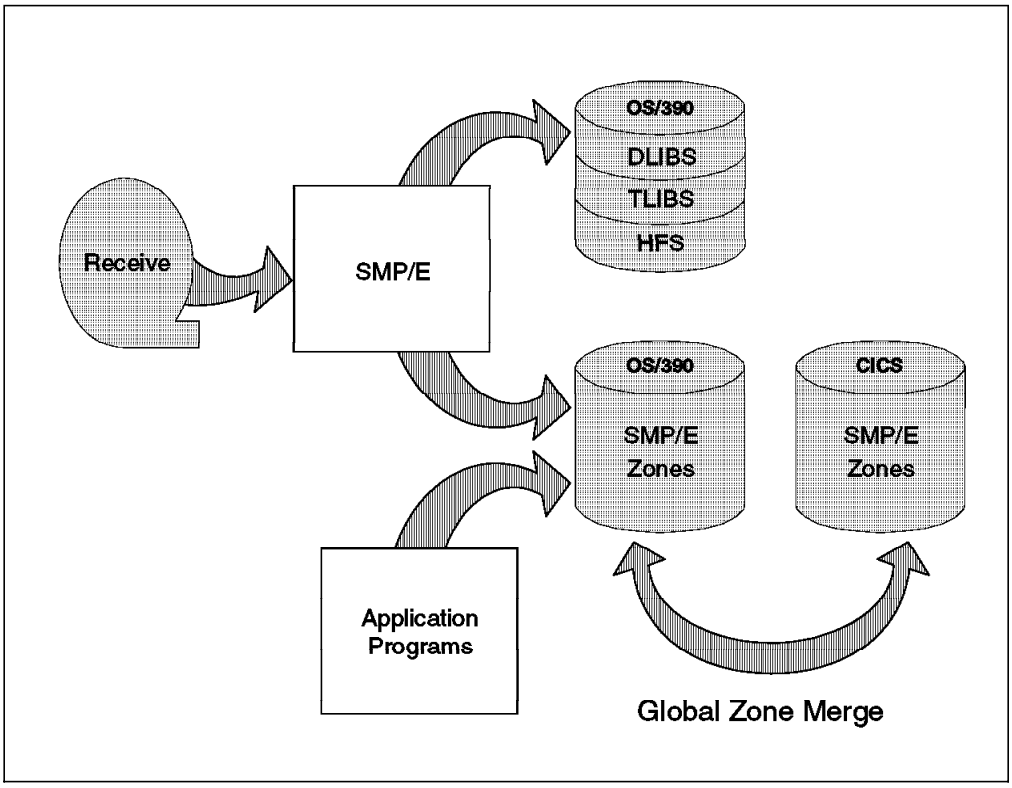


Figure 144. OS/390 Release 5 SMP/E Enhancements

7.1 SMPPTS Data Set Compaction

To reduce the space requirements of the SMPPTS data set, SMP/E is compacting PTF members within the data set during RECEIVE processing and expanding them during APPLY and ACCEPT processing. SMP/E is also providing a stand-alone service routine, GIMCPTS, which can be used to compact or expand PTFs outside of the RECEIVE, APPLY, and ACCEPT context. The advantage of the compression is that DASD space is reduced during PTF installation.

7.1.1 Impact on the SMP/E OPTIONS Entry

The OPTIONS entry defines processing options that are to be used for an SMP/E command or set of commands. One new subentry, Compact SMPPTS(COMPACT), is added to the OPTIONS entry to support SMPPTS dat set compaction. Compact SMPPTS specifies whether inline element data within SYSMODs in the SMPPTS data set should be compacted. The element data is normally compacted during the RECEIVE and GZONEMERGE commands.

The UCL operand is COMPACT(YES|NO).

YES YES indicates inline element data within SYSMODs in SMPPTS should be compacted to reduce the space requirements of SMP during the RECEIVE and GZONEMERGE commands. The element data is expanded as needed during APPLY and ACCEPT command processing. YES is the default.

NO NO indicates inline element data within SYSMODs in SMPPTS should not be compacted during the RECEIVE and GZONEMERGE commands. The element data will reside in the SMPPTS data set in its original form.

7.1.2 Impact on the SMP/E RECEIVE Command

The RECEIVE command is the first SMP/E command used to process any SYSMOD. It reads data from tape files or DASD data sets, writes entries into the global zone and members into the SMPPTS data set, and copies temporary libraries (SMPTLIBs) for later SMP/E processing. The RECEIVE command processes SYSMODs contained in the SMPPTFIN data set. The SYSMODs in SMPPTFIN are made up of Modification Control Statements (MCS), and may also contain inline element data. The RECEIVE command reads the SYSMODs from the SMPPTFIN data set and writes them to members in the SMPPTS data set.

SMPPTS contains one member for each SYSMOD that has been received. Currently, the member within SMPPTS is an exact copy of the SYSMOD found in the SMPPTFIN data set. To reduce the size of each member, and therefore reduce the space requirements for the entire SMPPTS data set, RECEIVE command processing will be changed to automatically compact inline element data within SYSMODs when writing members to the SMPPTS data set. This will be the case whenever the new Compact SMPPTS subentry in the active OPTIONS entry indicates compaction is to be performed (this is the default), and the driving system is at least MVS/ESA 4.3, which supports compression and expansion services. If the new Compact SMPPTS subentry in the active OPTIONS entry indicates no compaction is to occur, or the driving operating system does not support compression and expansion services, then no compaction of SMPPTS members is performed.

7.1.3 Impact on the SMP/E APPLY and ACCEPT Commands

The APPLY command is used to install elements from SYSMODs into the target libraries controlled by the target zone specified on the preceding SET command. Likewise, the ACCEPT command is used to install elements from SYSMODs into the distribution libraries controlled by the distribution zone specified on the preceding SET command. For both APPLY and ACCEPT the SYSMODs to be processed reside in the global zone and as members in the SMPPTS data set.

The members within the SMPPTS contain the Modification Control Statement (MCS) to define a SYSMOD, and may also contain inline element data. This data is the actual elements to be installed by SMP/E into the target and distribution libraries during APPLY and ACCEPT processing (object modules, assembler source, and so on).

In preparation for installation of these elements, the inline data is copied from the SYSMOD member in the SMPPTS data set to members of SMPWRKn data sets.

The members of the SMPWRKn data sets are used as input when invoking the system utilities to update the target and distribution libraries during APPLY and ACCEPT processing. Since the inline element data may now reside in the SMPPTS member in a compacted state, the data must therefore be expanded before it is installed into the target and distribution libraries during APPLY and ACCEPT processing.

7.1.4 Impact on the SMP/E GZONEMERGE Command

The GZONEMERGE command is used to merge information from one global zone and SMPPTS data set into another global zone and SMPPTS data set. To support reducing the space requirements of SMPPTS data sets, the GZONEMERGE command will be changed to automatically compact inline data within SYSMODs when copying members to an SMPPTS data set. This will be the case whenever the new Compact SMPPTS subentry in the active OPTIONS entry indicates compaction is to be performed (this is the default), and the driving system is at least MVS/ESA 4.3, which supports compression and expansion services. If the new Compact SMPPTS subentry in the active OPTIONS entry indicates no compaction is to occur, or the driving operating system does not support compression and expansion services, then no compaction of SMPPTS members is performed.

7.1.5 Impact on the SMP/E LIST Command

The LIST command is used to obtain formatted listings of SMP/E entries. LIST command processing is modified to format the new Compact SMPPTS subentry in the OPTIONS entry.

7.1.6 Impact on the SMP/E UCLIN Command

The UCLIN command is used to create or modify SMP/E entries. UCLIN command processing is modified to support the new Compact SMPPTS subentry in Global zone OPTIONS entries. Specifically, the new Compact SMPPTS subentry is supported by the ADD, REP and DEL operands of UCLIN.

7.1.7 Impact on the SMP/E Dialogs

To allow a user to view the original, uncompacted format of SYSMODS in the SMPPTS data set, the SMP/E dialogs were updated to view MCS entries in the global zone. The MCS entries do not reside in the Global zone, but rather correspond to SMPPTS data set members. When MCS entries are viewed, SMP/E will expand inline element data if it has been compacted.

In addition, the new Compact SMPPTS subentry will be displayed when viewing OPTIONS entries in the Global zone as shown in Figure 145 on page 168.

```
                          CSI QUERY - OPTIONS ENTRY
===>

 To return to the previous panel, enter END .
 NOTE: ENTER  R  ON COMMAND LINE TO DISPLAY RECOVERY OPTIONS


 Entry Type:  OPTIONS                              Zone Name: GLOBAL
 Entry Name:  OS5OPT                               Zone Type: GLOBAL

                         -------  Utilities  -------
   AMS:   AMS                LKED:    LKED          COPY:    COPY
   ASM:   ASMA90             UPDATE:  UPDATE        S/ZAP:   ZAP
   COMP:  COMP               HFSCOPY:               IOSUP:

                         -------   Options   -------
   NUCID:     6              PAGELEN:   66          PEMAX:
   PURGE:     YES            REJECT:    NO          MSGWIDTH:  80
   SAVEMTS:   NO             SAVESTS:   NO          MSGFILTER:
   CHANGEFILE:YES            COMPACT:   YES

 DSPREFIX: OS390CB.R5CB02
   PRIM:      800             SEC:      200           DIR:     1500
```

Figure 145. CSI Query - Options Entry

7.1.8 Compaction Service Routine

The new GIMCPTS service routine is used to compact or expand inline element data within SYSMODs. A user may compact SYSMOD data within an existing large SMPPTS data set. After processing, the total space requirement of the SMPPTS data set will be reduced.

7.1.9 Implementation of SMPPTS Data Set Compaction

To be able to implement SMPPTS data set compaction, the following actions may be taken:

- Use the GIMCPTS service routine to compact the current SMPPTS data set.
- Change the OPTIONS COMPACT SMPPTS subentry to YES.

The sample JCL shown in Figure 146 was used to compact the SMPPTS data set.

```
//RALPHRJ JOB (999,POK),CLASS=A,MSGCLASS=X,NOTIFY=&SYSUID
//STEP1   EXEC PGM=GIMCPTS,PARM='COMPACT'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD DSN=OS390CB.R5CB02.SMPPTS,DISP=SHR
//SYSUT2    DD DSN=RALPHR.R5CB02.SMPPTS,DISP=OLD
```

Figure 146. Sample JCL to Compact the SMPPTS

The DD SYSPRINT is used by GIMCPTS for messages.

The DD SYSUT1 points to a sequential or partitioned input data set.

The DD SYSUT2 points to a sequential or partitioned output data set.

The complete syntax and description of the GIMCPTS service routine may be found in *OS/390 SMP/E Reference*, OS/390 SMP/E Reference.

To change the OPTIONS COMPACT SMPPTS subentry, two methods may be used: you could change the subentry by using UCLIN, or online by using the SMP/E dialogs. We changed the subentry by utilizing the dialogs.

7.2 Library Change Interface

The purpose of the Library Change Interface is to be able to provide a method of doing the following:

- Describing the results of APPLY and RESTORE command processing
- Identifying the libraries which changed during the APPLY and RESTORE
- Identifying members and aliases in those libraries that changed during the APPLY and RESTORE

7.2.1 Library Change Interface Data Sets

Different records describing the changes are written to new SMP/E data sets. The DDDEFs for the data sets are SMPDATA1 and SMPDATA2. The SMPDATA2 data set is used as a spill data set when SMPDATA1 becomes full. The attributes of the SMPDATA1 and SMPDATA2 data sets are as follows:

- They can only be defined to SMP/E with a DD statement or a DDDEF.
- They must be defined to each target zone to enable Library Change processing to occur for that target zone.
- Do not concatenate SMPDATA1/2 data sets.
- They may not be allocated as a path in the hierarchical file system.
- The size of the data set will vary depending on usage.
- The data set must be managed by the user.

The allocation attributes of the SMPDATA1/2 data set are as follows:

- Sequential data set with a BLKSIZE=296-32760, RECFM=VB DISP+MOD.
- If BLKSIZE is less than 296, SMP/E will use a default of 8800.
- DISP=MOD must be specified to maintain a cumulative history of SMP/E APPLY and RESTORE processing.
- If DISP other than MOD is used, then SMP/E will use MOD.
- An LRECL=292 will be used by SMP/E when allocating the data set.

7.2.2 Library Change Interface Record Types

The following record types are produced in either SMPDATA1 or SMPDATA2 when the Changefile subentry is set to YES.

Record Type	Description
A0	Alias Record Type 0
C0	Continuation Record Type 0

E0 Element Record Type 0

H0 Header Record Type 0

L0 Library Record Type 0

L1 Library Record Type 1

P0 SYSMOD Status Record Type 0

S0 SMP/E Status Record Type 0

T0 Trailer Record Type 0

For a detailed description of the records, refer to *OS/390 SMP/E Reference*, SC28-1806

7.2.3 Implementation of Library Change Interface

To implement the Library Change interface, the following steps were taken:

- Define the SMPDATA1/2 data sets for each Target Zone in SMP/E.
- Create DDDEF entries for each data set in each Target Zone.
- Change the target zone OPTIONS subentry CHANGEFILE to YES.

Once the above changes were made, we ran an SMP/E APPLY. Shown in the following figures are samples of the records written to SMPDATAn.

```
H0MVST1001998083210431000000000000000336000000000835
 ......
 ......
 ......
```

Figure 147. Sample Header Record Type 0

The purpose of the H0 record is to uniquely identify the start of a set of Library change records.

In Figure 147 the header record describes the following:

- MVST100 is the target zone name.
- 1998083 is the date that the APPLY completed.
- 210431 is the time that the APPLY completed for this change.
- No SYSMODS were applied in error.
- No SYSMODS were left in a status of INCMPLT.
- 336 SYSMODS were applied successfully.
- No SYSMODS were deleted.
- 835 SYSMODS were superseded in this APPLY.

```
POUQ06036APPLIED HMWL810PTF
 ......
 ......
 ......
POAN65602SUPD
 ......
 ......
 ......
```

Figure 148. Sample SYSMOD Status Record Type 0

Status records are created for SYSMODs for which some utility work was done during the command, and for SYSMODs superseded by SYSMODs that were successfully APPLIED. In Figure 148, the following is described:

- SYSMOD UQ06036 was successfully APPLIED.
- The status of the SYSMOD is APPLIED.
- The FMID of this SYSMOD is HMWL810.
- This SYSMOD is a PTF.
- SYSMOD AN65602 was superseded by another SYSMOD in this APPLY.

```
 ......
 ......
 ......
LOCBRDBRM               PDS      O35RS1SYS1.CBRDBRM
 ......
 ......
 ......
```

Figure 149. Sample Library Record Type 0

A single Library Record type 0 is created for each target library changed during APPLY or RESTORE processing that is not associated with a pathname. In Figure 149, the following is described:

- CBRDBRM is the SMP/E ddname associated with this record.
- The type of library associated with this record was a PDS.
- The volume associated with this library was O35RS1.
- The data set name was SYS1.CBRDBRM.

```
 ......
 ......
 ......
L1SAOPBIN        HFS     /service_r5/usr/lpp/Printsrv/bin/IBM/
 ......
 ......
 ......
```

Figure 150. Sample Library Record Type 1

A single Library Record type 1 is created for each target library changed during APPLY or RESTORE processing that is associated with a pathname. In Figure 150, the following is described:

- SAOPBIN is the ddname.
- The type of library associated with this ddname was an HFS library.
- The data set name associated with this ddname was /service_r5/usr/lpp/Printsrv/bin/IBM/.

```
 ......
 ......
 ......
EOADMACIN LMOD        ADDREP  SADMMOD
 ......
 ......
 ......
```

Figure 151. Sample Element Record Type 1

An element record type 0 is created for each element or LMOD that changed during APPLY or RESTORE. In Figure 151, the following is described:

- ADMACIN is the name of the SMP/E element or lmod processed.
- LMOD is the type of element.
- The action taken for this element was ADDREP.
- SADMMOD is the ddname where ADMACIN was added or replaced.

```
 ......
 ......
 ......
AOAFHPRNAGLMOD        ADDREP  SCEERUN CEEEV007
 ......
 ......
 ......
```

Figure 152. Sample Element Record Type 1

An alias record type 0 is created for each alias for an element or LMOD processed during APPLY or RESTORE processing. In Figure 152, the following is described:

- AFHPRNAG is the name of the SMP/E element or lmod processed.
- LMOD is the type of element.
- The action taken for this element was ADDREP.
- SCEERUN is the ddname where AFHPRNAG was added or replaced.

7.3 Enhanced RECEIVE Processing

Management of the SMPPTS data set has become a problem due to the size of the SMPPTS data set. The SMPPTS data set is also limited to a single DASD volume. Currently once a SYSMOD has been applied or accepted, you can remove a SYSMOD from the global zone and the SMPPTS data set using a couple of alternatives. Unfortunately, in the current situation, the SYSMOD can be received back into the global zone and the SMPPTS data set. With the enhanced RECEIVE processing, it is now possible to prevent this from occurring.

The following is a summary of the changes in SMP/E to cater for enhanced RECEIVE processing.

- The OPTIONS entry has two new subentries, Receive Zone Group and Receive Exclude Group.
- The RECEIVE command was changed to have two new BYPASS suboperands, APPLYCHECK and ACCEPTCHECK, as well as a new operand, ZONEGROUP.
- UCLIN was changed to be able to add, replace, and delete the new OPTIONS subentries.
- The List command was changed to display the new OPTIONS subentries.
- Various SMP/E dialogs were changed to cater for the new OPTIONS subentries.

For the complete syntax and the changes made to the SMP/E commands please refer to *OS/390 SMP/E Command Reference*, SC28-1805.

A Receive Zone group is a list of zones and/or zonesets eligible for APPLYCHECK and ACCEPTCHECK processing during RECEIVE command processing to determine whether the SYSMOD has been previously applied or accepted.

A Receive Exclude Zone Group is a list of zones and/or zonesets to be excluded during APPLYCHECK and ACCEPTCHECK processing during RECEIVE command processing to determine whether the SYSMOD has been previously applied or accepted.

7.3.1 Implementation of Enhanced Receive Processing

To be able to implement enhanced receive processing, the following actions may be taken:

- Define a Receive Zone Group to be used during SMP/E RECEIVE command processing. The definition can be done by either:
- Defining the zones via the Receive Zone Group subentry for an OPTIONS entry and making the OPTIONS entry active during the Receive or,
- Using the new ZONEGROUP operand on the RECEIVE command.

Figure 153 shows sample SMP/E statements for doing a RECEIVE with the ZONEGROUP operand.

```
SET    BOUNDARY (GLOBAL)
                .
RECEIVE
       SYSMODS
       ZONEGROUP(
                 TGT605
                )
              .
```

Figure 153. Sample RECEIVE Statement with ZONEGROUP Operand

In this sample, a RECEIVE of SYSMODS will be done. The ZONEGROUP operand specifies that the zone TGT605 will be checked and if a SYSMOD is applied or accepted in this zone, the SYSMOD will not be received.

Chapter 8. OS/390 HCD and OS/390 HCM Enhancements

The focus of HCD and HCM, in OS/390 Release 5, is to make it easier to configure and manage large amounts of data. In addition, HCM is available on Windows 95 and Windows NT.

8.1 HCD Enhancements

- HCD provides verification and priming of I/O Paths. This new function supports the priming of device self-description data, such as serial numbers and ESCON director port connections. It can be invoked as a separate step or when new configuration elements are defined.
  - The Verify I/O Configuration dialog from HCD retrieves the actual configuration from the system and compares it with the logical configuration from a production IODF. (This function is dependent on prereqs, see 8.2, “Verifying and Priming I/O Configurations Requirements” on page 176). The operational status and any discrepancies in the I/O paths are displayed. This function is intended to be applied before and after a full dynamic activate, to verify the system configuration against the active IODF configuration and against the configuration that is to be activated. This provides an additional check as to whether the configuration change matches the expectation.
  - A new function is added which allows the user to build or process the CONFIGXX PARMLIB member for the CHP and DEV statements. New members can be created or existing members updated.
  - The DISPLAY M=CONFIG(xx) command can be routed to a system in a sysplex and the results of the command routed back to the HCD dialog.
  - The Data Entry Field prompts are enhanced to allow priming of the processor, control unit, device and switch definitions from an active system (for example, serial numbers, VOLSERs and switch port names).
- HCD provides enhancements for large IODFs and distributed configurations. HCD also allows for the distribution of single configurations out of an IODF to a target system, and for the merging of distributed IODFs to a master IODF.

  The I/O configuration file (IODF) is the central data repository of HCD. While there may exist multiple IODFs, HCD in general only has one IODF in access and works with one IODF at a single point in time.

  An HCD user can choose among different IODF concepts which range from keeping a single IODF for the I/O definitions of the whole enterprise, up to keeping a separate IODF for each processor and OS configuration. Organizing and administering multiple IODFs is a complex task, especially for large customers.

  It is therefore recommended you have large IODFs that contain all the I/O configurations, for the following reasons:
  - Control units and devices that are shared between different processors are defined only once. This minimizes maintenance on IODFs.
  - For a full dynamic I/O reconfiguration, it is a requirement to have the IPLed OS configuration in the same production IODF as the processor configuration that has been selected for the active IOCDS.

- HCD is only able to detect CTC misconfigurations as reported in the CTC Connection report, when all the CTCs are defined in the same IODF.
- In defining CF connections, for each CFS channel path, the target CFR channel path has to be defined in the same IODF. If this is not done, the connection cannot be defined in HCD and the CFS control unit and CFS devices cannot be generated.
- To manage IOCDS data sets and IPL parameters within the CPCs of a S/390 microprocessor cluster from a focal-point HCD, the corresponding processor and OS configurations have to be defined in the same IODF.
- To dynamically reconfigure the I/O configuration of a system within a sysplex from a focal-point HCD, the systems (Processor and OS configurations) have to be defined in the same IODF.
- The scope of the textual and graphical report is a single IODF. In order to get a complete report, the I/O definitions of interest have to be kept in a single IODF.
- The scope of the validation function is a single IODF. In order to get the configuration validated over a broad range (for example, all objects that are connected via a switch), the I/O definitions must be in a single IODF.
- The scope of a physical configuration shown with HCM is a single IODF. In order to view a complete installation′s configuration, all the I/O definitions must reside in a single IODF.

These requirements lead to huge IODFs within large enterprises, which adds to the complexity in administering them. With the increased scope of I/O definitions, the required virtual space for the IODF increases. Due to the limitations of the user address space, the size of the IODF becomes more and more restricted. Relief for the primary user address space is provided by accessing the IODF data via a separate data space.

8.2 Verifying and Priming I/O Configurations Requirements

The verify and prime I/O configurations functions are based on the ESCON device self-description architecture. To get data from the active system, HCD uses the ESCON Manager or System Automation for OS/390 (I/O Operations) API for Query Services. This requires that ESCON Manager 1.3 (with APARs PN87285 and PN87286) or System Automation for OS/390 is installed and running on the same system as HCD.

To get the details of the I/O paths of a particular system in a sysplex, the following prereqs have to be met:

- A VTAM session between the local and the target system must exist.
- The target system must have ESCON Manager installed.

8.3 HCM Enhancements

- HCM Windows 95 and Windows NT Client Support

  The Windows 95 and Windows NT client support enables HCM to run on the Windows 95 and Windows NT platforms. This support is also available via SPE PTFs for HCM V.1.1.0 and OS/390 Version 2 Release 4 HCM. The

support eliminates the current restriction of HCM only being able to run on Windows 3.x and Win-OS/2.

- HCM Enhanced Filter Capabilities for HCM Diagrams

  This feature allows a configuration to be examined and searched by using a sophisticated, flexible, SQL-like, data viewing mechanism. Furthermore, HCM provides the ability to save and maintain a set of predefined filters for easy recall. Support is also available for OS/390 Version 2 Release 4 HCM via SPE PTFs. This enhancement makes it significantly easier to manage large amounts of configuration data.

8.3.1 APPC Setup for Windows NT

HCM running on Windows 95 or NT requires IBM Personal Communications (PCOMM) for Windows NT (Version 4.11 or later). The following procedure, customizes PCOMM for APPC to be used on a Windows 95/NT workstation. The current network is schematically shown in Figure 154.

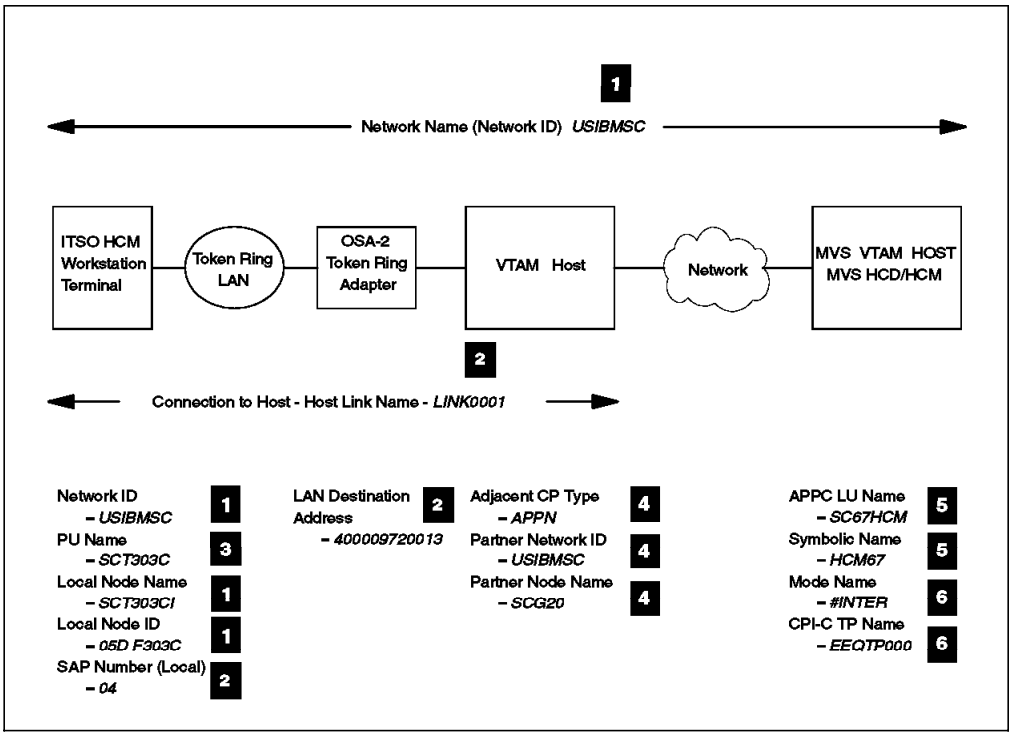


Figure 154. Example of the Network in an HCM Implementation

The first step in setting up the APPC configuration is to design the APPC network. Instructions for this step can be found in *Setting Up APPC Definitions of the HCM User's Guide*, SC33-6595. There are many different APPC implementations, of which the following is an example:

Notes for Figure 154:

1 Define the Node.
2 3 4 Define the LAN Connection.
5 Define a Partner LU 6.2.
6 Define CPI-C side information.

After installing PCOMM Version 4.11 or later, select ***SNA Node Configuration*** which can be found in the main folder Personal Communications. There are five

main options that need to be defined when setting up the APPC configuration, listed in the Node Configuration dialog. They are:

1. Configure Nodes
2. Configure Devices
3. Configure Connections
4. Configure Partner LU 6.2
5. Configure CPI-C Side Information

On the main dialog, Personal Communications SNA Node Configuration (Figure 155), select ***Configure Node*** and then select ***New***.

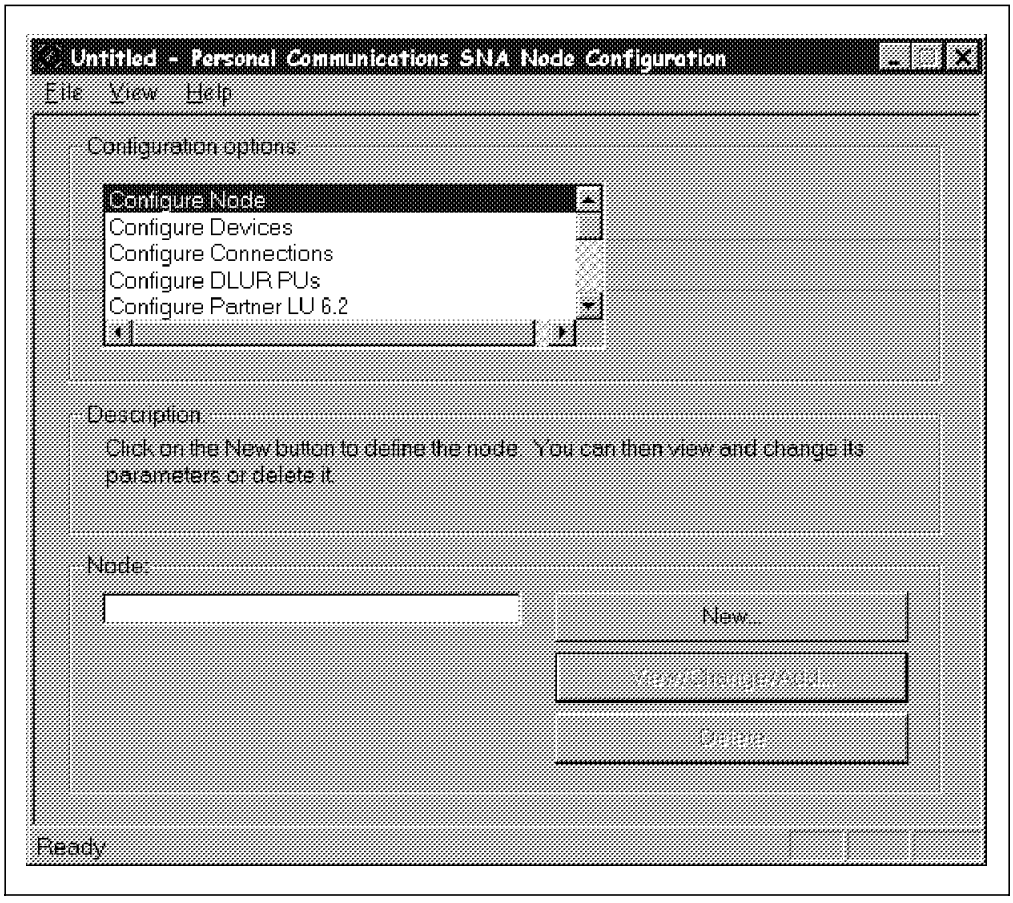


Figure 155. Personal Communications SNA Node Configuration Dialog

When the Define Node dialog (Figure 156 on page 179) is displayed, enter the fully qualified CP name which consists of the Network ID and the LU Name of the workstation. Enter the name of an Alias, (the value entered in the LU Name field can be used here). The next field to update is the Local Node ID field. Accept the default values on the Advanced and DLU Requester tabs of the dialog. When all the values have been entered, select ***OK***.

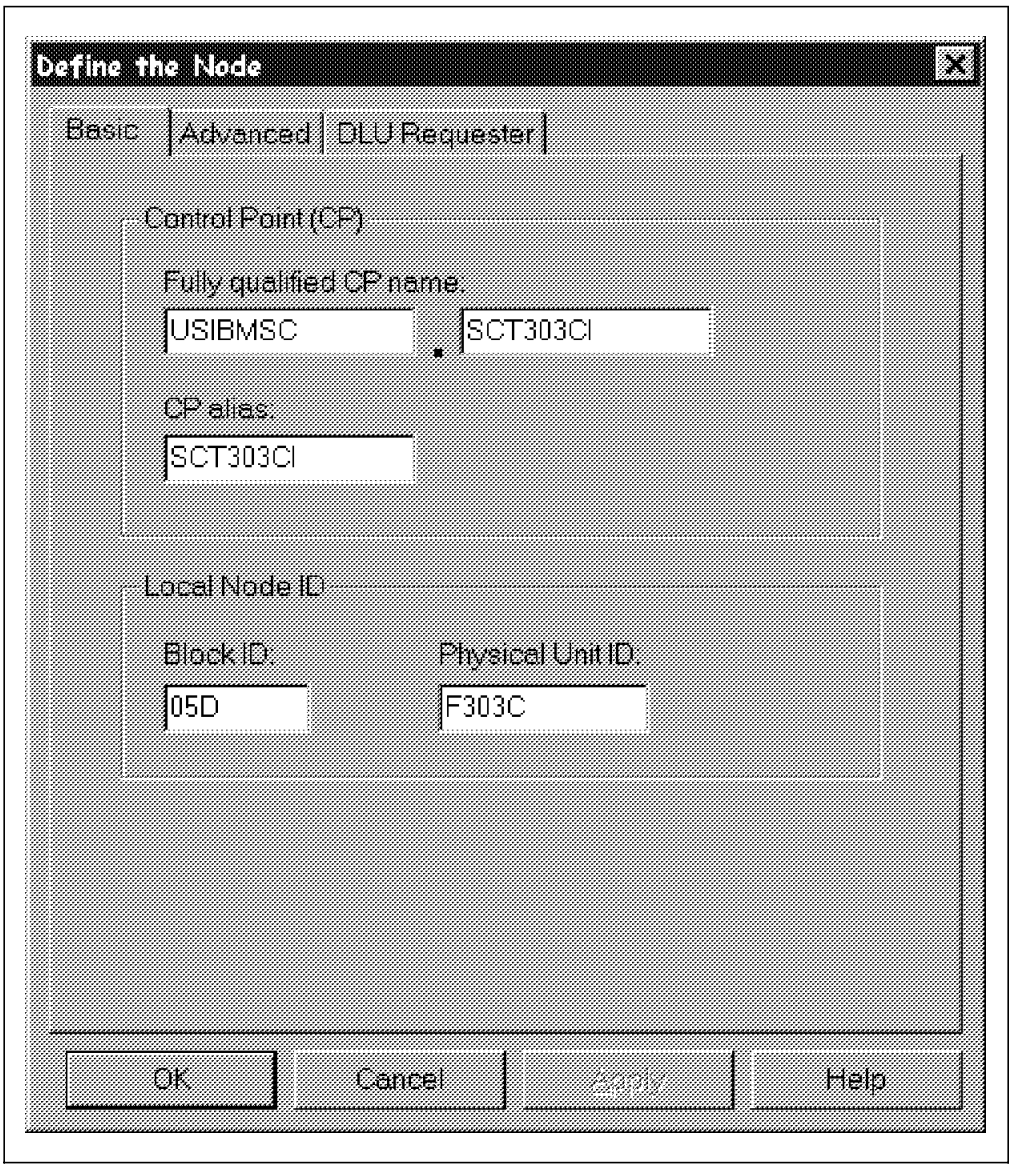


Figure 156. Personal Communications Define Node - Basic

After returning to the main dialog, select Configure Devices and then select ***New*** (Figure 157 on page 180). When the Define a LAN Device dialog is displayed, accept the defaults and select ***OK***.

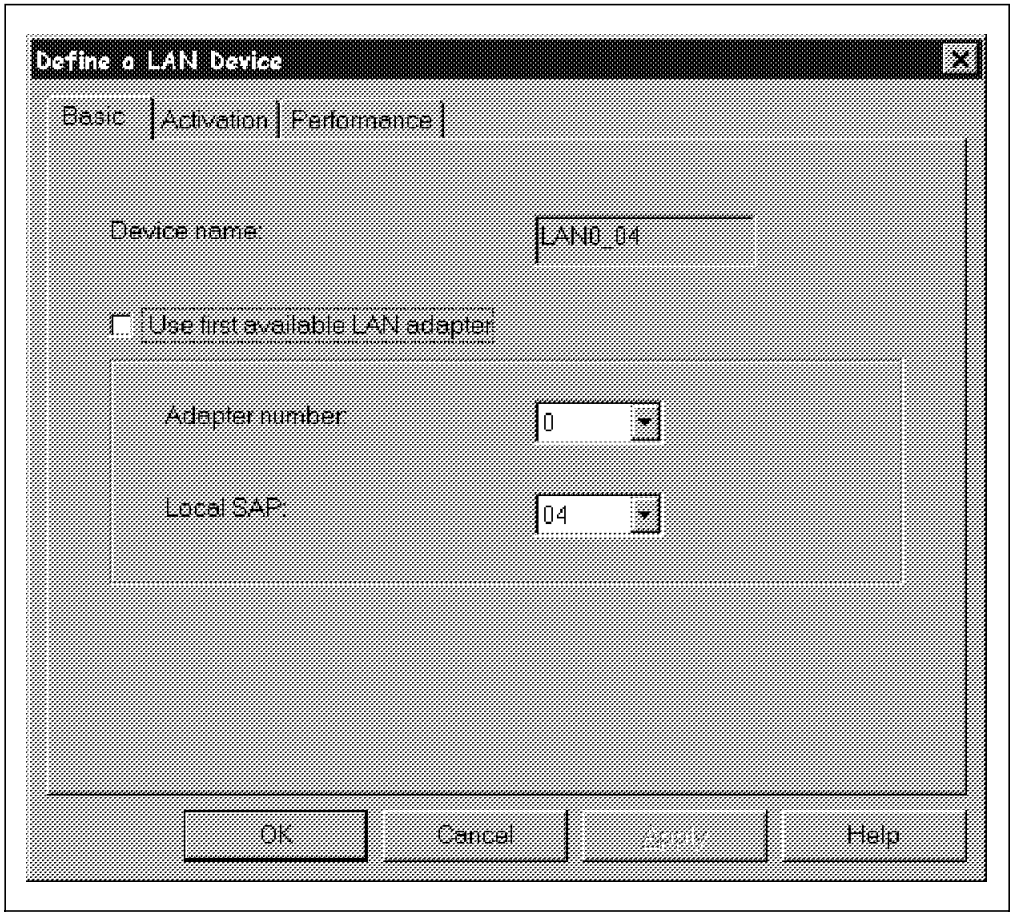


Figure 157. Personal Communications Define LAN Device - Basic

After returning to the main dialog, select ***Configure connections*** and then select ***New***.

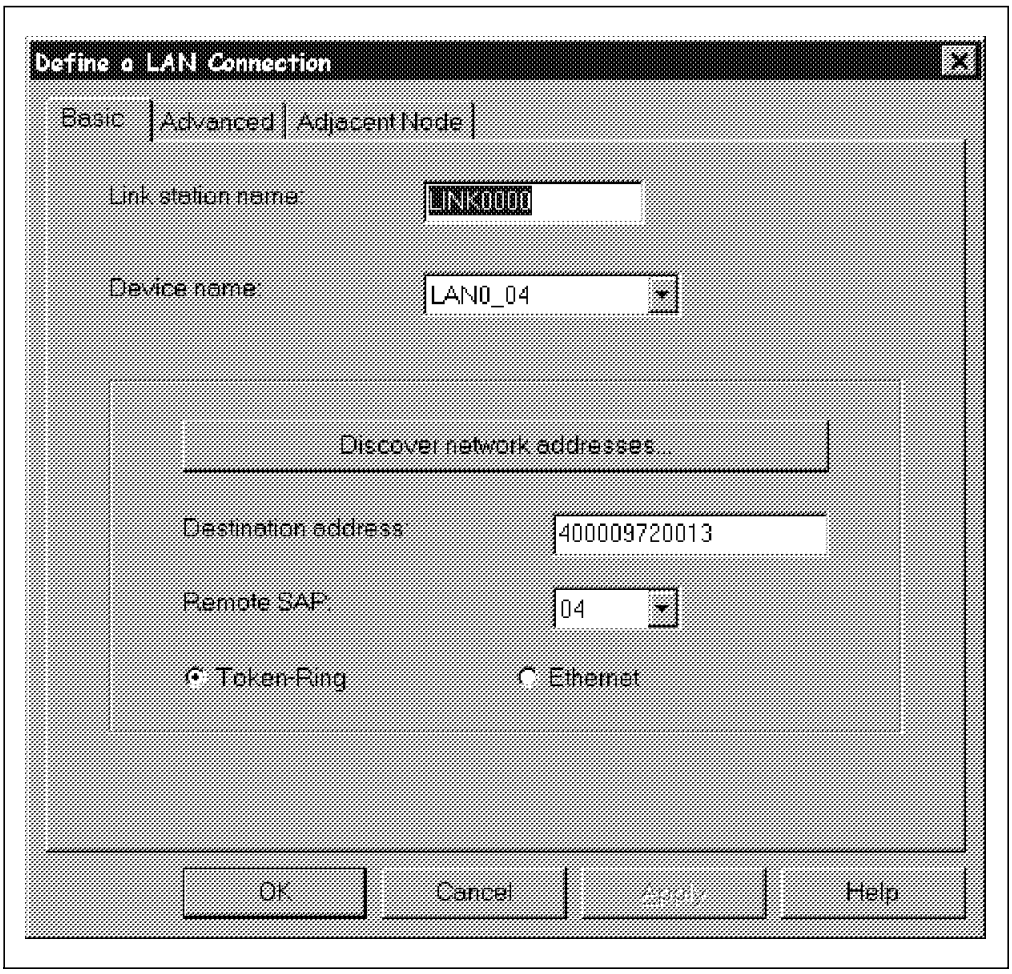


Figure 158. Personal Communications Define LAN Connection - Basic

When the Define a LAN Connection dialog (see Figure 158) is displayed, enter a Link Station Name or accept the default. Select the correct Device name related to the adapter installed in the workstation. Update the Destination address field with the MAC address of the network with which this workstation is establishing a session. When all the values have been entered, select the ***Advanced*** tab at the top to display the dialog shown in Figure 159 on page 182.

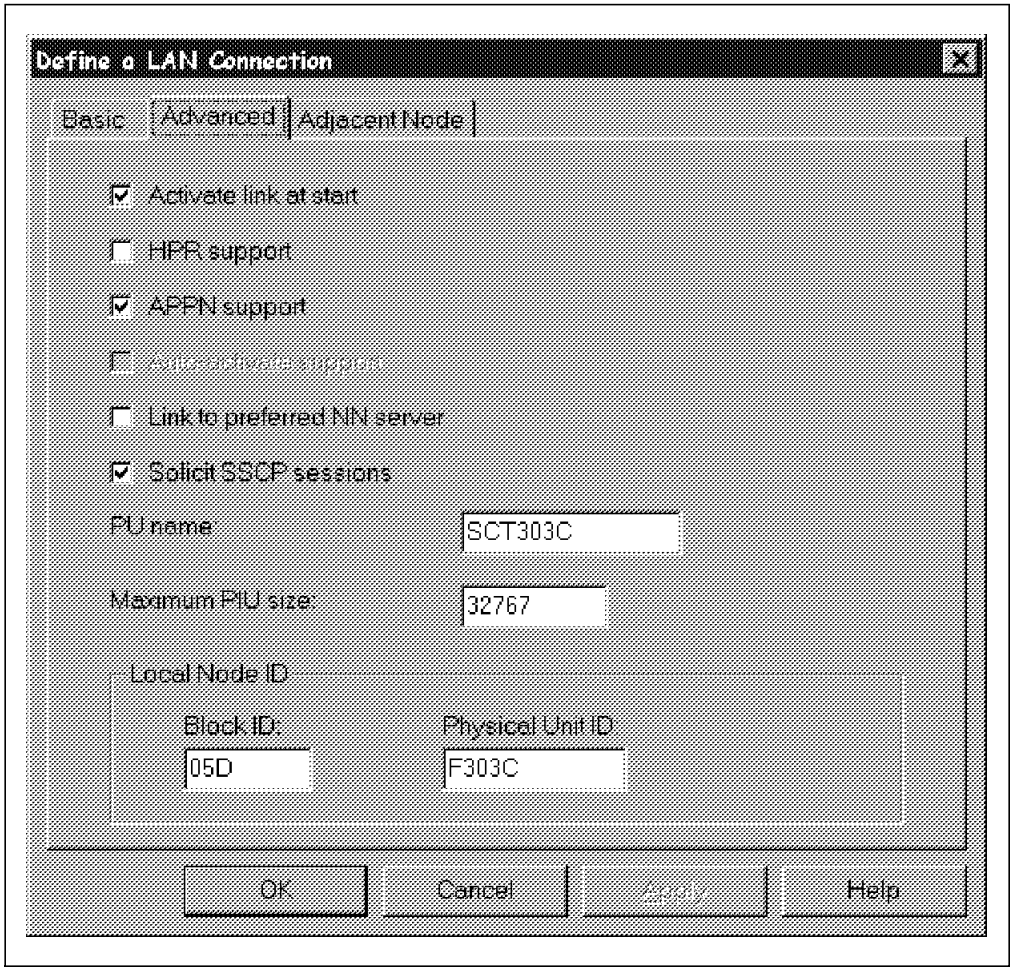


Figure 159. Personal Communications Define LAN Connection - Advanced

Select ***Activate link at start, APPN support*** and ***Solicit SSCP sessions***. Enter the PU address of this workstation in the PU name field. Accept the values in the remaining fields on this dialog. When all the values have been entered, select the ***Adjacent Node*** tab at the top to display the dialog shown in Figure 160 on page 183.

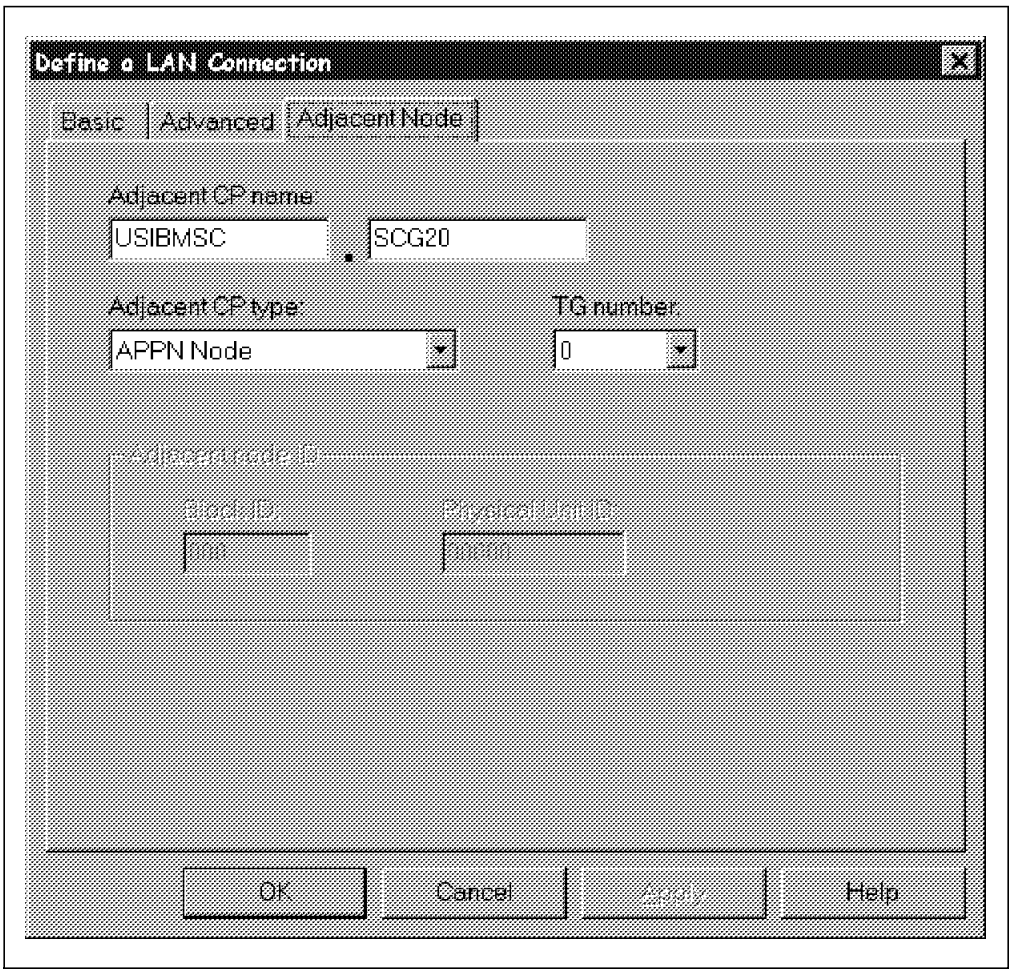


Figure 160. Personal Communications Define LAN Connection - Adjacent Node

Enter the Adjacent Control Point (CP) name, which consists of two parts, the Network Name and the CP name, concatenated by a period. This name is the name of the Control Point that is directly connected to the workstation across this link. Select ***APPN Node*** as the Adjacent CP Type. When all the values have been entered, select ***OK***.

After returning to the main dialog (Personal Communications SNA Node Configuration), select ***Configure Partner LU 6.2*** and then select ***New***.

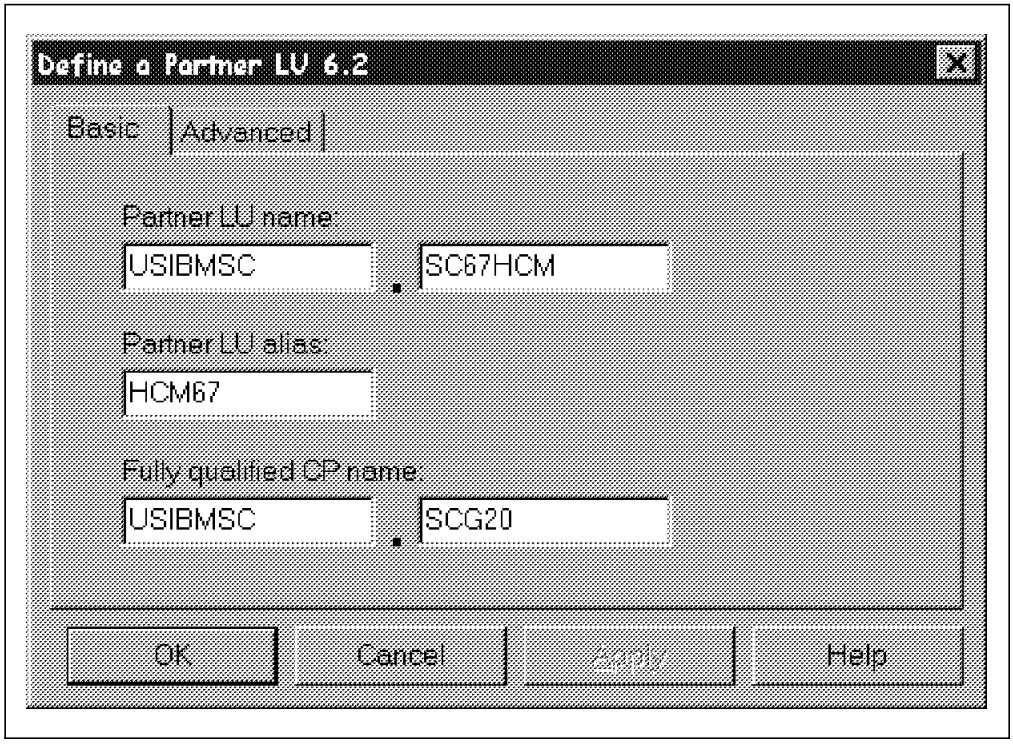


Figure 161. Personal Communications Define Partner LU - Basic

The Define Partner LU dialog (see Figure 161) is displayed. The Partner LU name is the name of the LU where the partner program is located. Enter the Partner LU name, which consists of two parts, the Network Name and the Partner LU name concatenated by a period. Enter a name in the Partner LU alias that corresponds with the CPI-C information entered in the next step. Enter the Fully qualified CP name of the owning network. When all the values have been entered, select ***OK***.

After returning to the main dialog, select ***Configure CPI-C side information*** and then select ***New***.

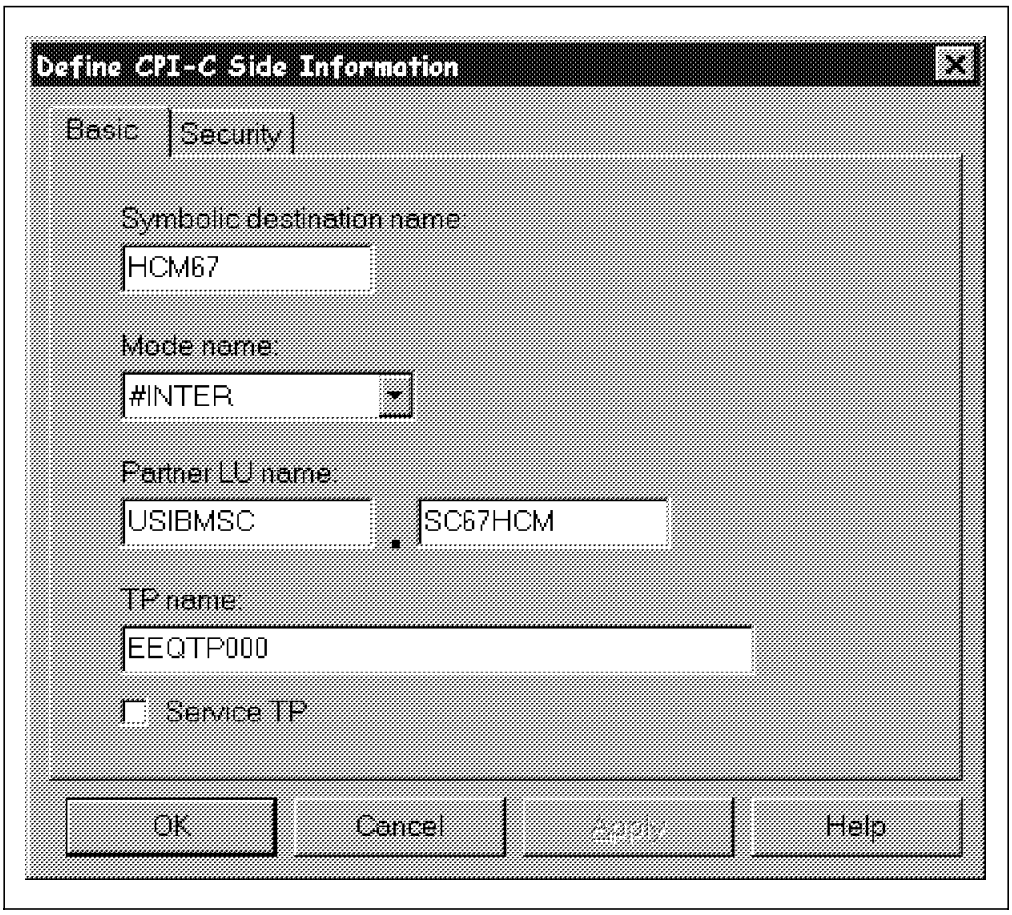


Figure 162. Personal Communications Define CPI-C - Basic

The ***Define CPI-C - Basic*** dialog (see Figure 162) is displayed. Enter the Symbolic destination name which corresponds with the Partner LU name (Figure 161 on page 184). Select **#INTER** in the Mode Name field, enter the Partner LU name and the TP name as defined in the APPC/MVS setup. When all the values have been entered, select ***OK***.

After returning to the main dialog, save the APPC definitions in a file. Start the PCOMM APPC application by opening the file that has been saved in the previous step from the SNA Node Operation dialog.

APPC has to be started before the HCM can be invoked.

Appendix A. RFC 1179

Network Printing Working Group — L. McLaughlin III, Editor
Request for Comments: 1179 — The Wollongong Group
August 1990

Line Printer Daemon Protocol

Status of this Memo

This RFC describes an existing print server protocol widely used on the Internet for communicating between line printer daemons (both clients and servers). This memo is for informational purposes only, and does not specify an Internet standard. Please refer to the current edition of the "IAB Official Protocol Standards" for the standardization state and status of this protocol. Distribution of this memo is unlimited.

1. Introduction

The Berkeley versions of the Unix(tm) operating system provide line printer spooling with a collection of programs: lpr (assign to queue), lpq (display the queue), lprm (remove from queue), and lpc (control the queue). These programs interact with an autonomous process called the line printer daemon. This RFC describes the protocols with which a line printer daemon client may control printing.

This memo is based almost entirely on the work of Robert Knight at Princeton University. I gratefully acknowledge his efforts in deciphering the UNIX lpr protocol and producing earlier versions of this document.

2. Model of Printing Environment

A group of hosts request services from a line printer daemon process running on a host. The services provided by the process are related to printing jobs. A printing job produces output from one file. Each job will have a unique job number which is between 0 and 999, inclusive. The jobs are requested by users, which have names. These user names may not start with a digit.

3. Specification of the Protocol

The specification includes file formats for the control and data files as well as messages used by the protocol.

3.1 Message formats

LPR is a TCP-based protocol. The port on which a line printer daemon listens is 515. The source port must be in the range 721 to 731, inclusive. A line printer daemon responds to commands sent to its port. All commands begin with a single octet code, which is a binary number that represents the requested function. The code is immediately followed by the ASCII name of the printer queue name on which the function is to be performed. If there are other operands to the command, they are separated from the printer queue name with white space (ASCII space, horizontal tab, vertical tab, and form feed). The end of the command is indicated with an ASCII line feed character.

4. Diagram Conventions

The diagrams in the rest of this RFC use these conventions. These diagrams show the format of an octet stream sent to the server. The outermost box represents this stream. Each box within the outermost one shows one portion of the stream. If the contents of the box are two decimal digits, this indicates that the binary 8 bit value is to be used. If the contents are two uppercase letters, this indicates that the corresponding ASCII control character is to be used. An exception to this is that the character SP can be interpreted as

white space. (See the preceding section for a definition.) If the contents is a single letter, the ASCII code for this letter must be sent. Otherwise, the contents are intended to be mnemonic of the contents of the field which is a sequence of octets.

5. Daemon commands

The verbs in the command names should be interpreted as statements made to the daemon. Thus, the command "Print any waiting jobs" is an imperative to the line printer daemon to which it is sent. A new connection must be made for each command to be given to the daemon.

5.1 O1 - Print any waiting jobs

```
+----+-------+----+
  O1   Queue   LF
+----+-------+----+
```

Command code - 1

Operand - Printer queue name

This command starts the printing process if it not already running.

5.2 O2 - Receive a printer job

```
+----+-------+----+
  O2   Queue   LF
+----+-------+----+
```

Command code - 2

Operand - Printer queue name

Receiving a job is controlled by a second level of commands. The daemon is given commands by sending them over the same connection. The commands are described in the next section (6).

After this command is sent, the client must read an acknowledgement octet from the daemon. A positive acknowledgement is an octet of zero bits. A negative acknowledgement is an octet of any other pattern.

5.3 O3 - Send queue state (short)

```
+----+-------+----+------+----+
  O3   Queue   SP   List   LF
+----+-------+----+------+----+
```

Command code - 3

Operand 1 - Printer queue name

Other operands - User names or job numbers

If the user names or job numbers or both are supplied, then only thos jobs for those users or with those numbers will be sent.

The response is an ASCII stream that describes the printer queue. The stream continues until the connection closes. Ends of lines are indicated with ASCII LF control characters. The lines may also contain ASCII HT control characters.

5.4 O4 - Send queue state (long)

```
+----+-------+----+------+----+
  O4   Queue   SP   List   LF
+----+-------+----+------+----+
```

Command code - 4

Operand 1 - Printer queue name

Other operands - User names or job numbers

If the user names or job numbers or both are supplied then only those jobs for those users or with those numbers will be sent.

The response is an ASCII stream that describes the printer queue. The stream continues until the connection closes. Ends of lines are indicated with ASCII LF control characters. The lines may also contain ASCII HT control characters.

5.5 O5 - Remove jobs

```
+----+-------+----+-------+----+------+----+
  O5   Queue   SP   Agent   SP   List   LF
+----+-------+----+-------+----+------+----+
```

Command code - 5
Operand 1 - Printer queue name
Operand 2 - User name making request (the agent)
Other operands - User names or job numbers

This command deletes the print jobs from the specified queue that are listed as the other operands. If only the agent is given, the command is to delete the currently active job. Unless the agent is "root", it is not possible to delete a job which is not owned by the user. This is also the case for specifying user names instead of numbers. That is, agent "root" can delete jobs by user name but no other agents can.

6. Receive job subcommands

These commands are processed when the line printer daemon has been given the receive job command. The daemon will continue to process commands until the connection is closed.

After a subcommand is sent, the client must wait for an acknowledgement from the daemon. A positive acknowledgement is an octet of zero bits. A negative acknowledgement is an octet of any other pattern.

LPR clients SHOULD be able to send the receive data file and receive control file subcommands in either order. LPR servers MUST be able to receive the control file subcommand first and SHOULD be able to receive the data file subcommand first.

6.1 01 - Abort job

Command code - 1

```
+----+----+
  01   LF
+----+----+
```

No operands should be supplied. This subcommand will remove any files that have been created during this "Receive job" command.

6.2 02 - Receive control file

```
+----+-------+----+------+----+
  02   Count   SP   Name   LF
+----+-------+----+------+----+
```

Command code - 2
Operand 1 - Number of bytes in control file
Operand 2 - Name of control file

The control file must be an ASCII stream with the ends of lines indicated by ASCII LF. The total number of bytes in the stream is sent as the first operand. The name of the control file is sent as the second. It should start with ASCII "cfA", followed by a three digit job number, followed by the host name which has constructed the control file. Acknowledgement processing must occur as usual after the command is sent.

The next "Operand 1" octets over the same TCP connection are the intended contents of the control file. Once all of the contents have been delivered, an octet of zero bits is sent as an indication that the file being sent is complete. A second level of acknowledgement processing must occur at this point.

6.3 03 - Receive data file

```
+----+-------+----+------+----+
  03   Count   SP   Name   LF
+----+-------+----+------+----+
```

Command code - 3
Operand 1 - Number of bytes in data file
Operand 2 - Name of data file

The data file may contain any 8-bit values at all. The total number of bytes in the stream may be sent as the first operand, otherwise the field should be cleared to 0. The name of the data file should

start with ASCII "dfA". This should be followed by a 3-digit job number. The job number should be followed by the host name that has constructed the data file. Interpretation of the contents of the data file is determined by the contents of the corresponding control file. If a data file length has been specified, the next "Operand 1" octets over the same TCP connection are the intended contents of the data file. In this case, once all of the contents have been delivered, an octet of zero bits is sent as an indication that the file being sent is complete. A second level of acknowledgement processing must occur at this point.

7. Control file lines

This section discusses the format of the lines in the control file that are sent to the line printer daemon.

Each line of the control file consists of a single, printable ASCII character that represents a function to be performed when the file is printed. Interpretation of these command characters is case-sensitive. The rest of the line after the command character is the command's operand. No leading white space is permitted after the command character. The line ends with an ASCII new line.

Those commands which have a lowercase letter as a command code are used to specify an actual printing request. The commands that use uppercase are used to describe parametric values or background conditions.

Some commands must be included in every control file. These are 'H' (responsible host) and 'P' (responsible user). Additionally, there must be at least one lowercase command to produce any output.

7.1 C - Class for banner page

```
+---+-------+----+
  C   Class   LF
+---+-------+----+
```

Command code - 'C'

Operand - Name of class for banner pages

This command sets the class name to be printed on the banner page. The name must be 31 or fewer octets. The name can be omitted. If it is, the name of the host on which the file is printed will be used. The class is conventionally used to display the host from which the printing job originated. It will be ignored unless the print banner command ('L') is also used.

7.2 H - Host name

```
+---+------+----+
  H   Host   LF
+---+------+----+
```

Command code - 'H'

Operand - Name of host

This command specifies the name of the host which is to be treated as the source of the print job. The command must be included in the control file. The name of the host must be 31 or fewer octets.

7.3 I - Indent Printing

```
+---+-------+----+
  I   count   LF
+---+-------+----+
```

Command code - 'I'

Operand - Indenting count

This command specifies that, for files that are printed with the 'f', of columns given. (It is ignored for other output generating commands.) The identing count operand must be all decimal digits.

7.4 J - Job name for banner page

```
+---+----------+----+
  J   Job name   LF
```

```
      +---+----------+----+
      Command code - 'J'
      Operand - Job name
```
This command sets the job name to be printed on the banner page. The name of the job must be 99 or fewer octets. It can be omitted. The job name is conventionally used to display the name of the file or files which were "printed". It will be ignored unless the print banner command ('L') is also used.

7.5 L - Print banner page

```
      +---+------+----+
        L   User   LF
      +---+------+----+
      Command code - 'L'
      Operand - Name of user for burst pages
```
This command causes the banner page to be printed. The user name can be omitted. The class name for banner page and job name for banner page commands must precede this command in the control file to be effective.

7.6 M - Mail When Printed

```
      +---+------+----+
        M   user   LF
      +---+------+----+
      Command code - 'M'
      Operand - User name
```
This entry causes mail to be sent to the user given as the operand at the host specified by the 'H' entry when the printing operation ends (successfully or unsuccessfully).

7.7 N - Name of source file

```
      +---+------+----+
        N   Name   LF
      +---+------+----+
      Command code - 'N'
      Operand - File name
```
This command specifies the name of the file from which the data file was constructed. It is returned on a query and used in printing with the 'p' command when no title has been given. It must be 131 or fewer octets.

7.8 P - User identification

```
      +---+------+----+
        P   Name   LF
      +---+------+----+
      Command code - 'P'
      Operand - User id
```
This command specifies the user identification of the entity requesting the printing job. This command must be included in the control file. The user identification must be 31 or fewer octets.

7.9 S - Symbolic link data

```
      +---+--------+----+-------+----+
        S   device   SP   inode   LF
      +---+--------+----+-------+----+
      Command code - 'S'
      Operand 1 - Device number
      Operand 2 - Inode number
```
This command is used to record symbolic link data on a Unix system so that changing a file's directory entry after a file is printed will not print the new file. It is ignored if the data file is not symbolically linked.

7.10 T - Title for pr

```
      +---+-------+----+
```

```
        T   title   LF
      +---+-------+----+
      Command code - 'T'
      Operand - Title text
   This command provides a title for a file which is to be printed with
   either the 'p' command.  (It is ignored by all of the other printing
   commands.)  The title must be 79 or fewer octets.
7.11 U - Unlink data file
      +---+------+----+
        U   file   LF
      +---+------+----+
      Command code - 'U'
      Operand - File to unlink
   This command indicates that the specified
   file is no longer needed.
   This should only be used for data files.
7.12 W - Width of output
      +---+-------+----+
        W   width   LF
      +---+-------+----+
      Command code - 'W'
      Operand - Width count
   This command limits the output to the specified number of columns for
   the 'f', 'l', and 'p' commands.  (It is ignored for other output
   generating commands.)  The width count operand must be all decimal
   digits.  It may be silently reduced to some lower value.  The default
   value for the width is 132.
7.13 1 - troff R font
      +---+------+----+
        1   file   LF
      +---+------+----+
      Command code - '1'
      Operand - File name
   This command specifies the file name for the troff R font.   This
   is the font which is printed using Times Roman by default.
7.14 2 - troff I font
      +---+------+----+
        2   file   LF
      +---+------+----+
      Command code - '2'
      Operand - File name
   This command specifies the file name for the troff I font.   This
   is the font which is printed using Times Italic by default.
7.15 3 - troff B font
      +---+------+----+
        3   file   LF
      +---+------+----+
      Command code - '3'
      Operand - File name
   This command specifies the file name for the troff B font.   This
   is the font which is printed using Times Bold by default.
7.16 4 - troff S font
      +---+------+----+
        4   file   LF
      +---+------+----+
      Command code - '4'
      Operand - File name
   This command specifies the file name for the troff S font.   This
   is the font which is printed using Special Mathematical Font by
```

default.

7.17 c - Plot CIF file

```
+---+------+----+
  c   file   LF
+---+------+----+
```

Command code - 'c'

Operand - File to plot

This command causes the data file to be plotted, treating the data as CIF (CalTech Intermediate Form) graphics language.

7.18 d - Print DVI file

```
+---+------+----+
  d   file   LF
+---+------+----+
```

Command code - 'd'

Operand - File to print

This command causes the data file to be printed, treating the data as DVI (TeX output).

7.19 f - Print formatted file

```
+---+------+----+
  f   file   LF
+---+------+----+
```

Command code - 'f'

Operand - File to print

This command causes the data file to be printed as a plain text file, providing page breaks as necessary. Any ASCII control characters which are not in the following list are discarded: HT, CR, FF, LF, and BS.

7.20 g - Plot file

```
+---+------+----+
  g   file   LF
+---+------+----+
```

Command code - 'g'

Operand - File to plot

This command causes the data file to be plotted, treating the data as output from the Berkeley Unix plot library.

7.21 k - Reserved for use by Kerberized LPR clients and servers.

7.22 l - Print file leaving control characters

```
+---+------+----+
  l   file   LF
+---+------+----+
```

Command code - 'l' (lower case L)

Operand - File to print

This command causes the specified data file to be printed without filtering the control characters (as is done with the 'f' command).

7.23 n - Print ditroff output file

```
+---+------+----+
  n   file   LF
+---+------+----+
```

Command code - 'n'

Operand - File to print

This command prints the data file to be printed, treating the data as ditroff output.

7.24 o - Print Postscript output file

```
+---+------+----+
  o   file   LF
+---+------+----+
```

Command code - 'o'

Operand - File to print

This command prints the data file to be printed, treating the data as

standard Postscript input.

7.25 p - Print file with 'pr' format

```
+---+------+----+
  p   file   LF
+---+------+----+
Command code - 'p'
Operand - File to print
```

This command causes the data file to be printed with a heading, page numbers, and pagination. The heading should include the date and time that printing was started, the title, and a page number identifier followed by the page number. The title is the name of the file as specified by the 'N' command, unless the 'T' command (title) has been given. After a page of text has been printed, a new page is started with a new page number. (There is no way to specify the length of the page.)

7.26 r - File to print with FORTRAN carriage control

```
+---+------+----+
  r   file   LF
+---+------+----+
Command code - 'r'
Operand - File to print
```

This command causes the data file to be printed, interpreting the first column of each line as FORTRAN carriage control. The FORTRAN standard limits this to blank, "1", "0", and "+" carriage controls. Most FORTRAN programmers also expect "-" (triple space) to work as well.

7.27 t - Print troff output file

```
+---+------+----+
  t   file   LF
+---+------+----+
Command code - 't'
Operand - File to print
```

This command prints the data file as Graphic Systems C/A/T phototypesetter input. This is the standard output of the Unix "troff" command.

7.28 v - Print raster file

```
+---+------+----+
  v   file   LF
+---+------+----+
Command code - 'v'
Operand - File to print
```

This command prints a Sun raster format file.

7.29 z - Reserved for future use with the Palladium print system.

REFERENCES and BIBLIOGRAPHY

Computer Science Research Group, "UNIX Programmer's Reference Manual", USENIX, 1986.

Hon and Sequin, "A Guide to LSI Implementation", XEROX PARC, 1980.

Knuth, D., "TeX The Program".

Kernighan, B., "A Typesetter-independent TROFF".

"Model C/A/T Phototypesetter", Graphic Systems, Inc. Hudson, N.H.

Sun Microsystems, "Pixrect Reference Manual", Sun Microsystems, Mountain View, CA, 1988.

Security Considerations

Security issues are not discussed in this memo.

Author's Address

Leo J. McLaughlin III
The Wollongong Group
1129 San Antonio Road

Palo Alto, CA 94303
Phone: 415-962-7100
EMail: ljm@twg.com

Appendix B. Permission Bits

You control access to a file and directory that you own through its permission bits. The permission bits are often called the mode. You can set or change permissions for your file and directories. To change permissions, you must be the owner or a superuser. You can specify the mode in symbolic form or as an octal value.

There are three classes of users whose access you can control:

Owner The owner of the file or directory, whose UID matches the UID for the file

Group A member of the group whose GID matches the GID for the file

Other Anyone else

When you first create a file or directory, the system sets default read, write, and execute (rwx) permissions. For example, if you issue the mkdir shell command, the default permission bit settings are:

- owner=rwx
- group=rwx
- other=rwx

In octal form, the setting is 777.

To illustrate this point further, if you issue the mkdir command from TSO, the default settings are:

- owner=rwx
- group=r-x
- other=r-x

The octal form is 755.

B.1 Permission Bit Settings

Each position of a 4-bit setting indicates a different type of access:

- In position 1 are the bits that set permission for set-user-ID on access, set-group-ID on access, or the sticky bit. Specifying this position is optional.
- In position 2 are the bits that set permissions for the owner of the file. Specifying this position is required.
- In position 3 are the bits that set permissions for the group that the owner belongs to. Specifying this position is required.
- In position 4 are the bits that set permissions for others. Specifying this position is required.

The value for a setting is as follows:

0 Off

1 Sticky bit on

2 Set-group-ID on execution

3 Set-group-ID on execution and set the sticky bit on.

4 Set-user-ID on execution

5 Set-user-ID on execution and set the sticky bit on.

6 Set-user-ID and set-group-ID on execution

7 Set-user-ID and set-group-ID on execution and set the sticky bit on.

Appendix C. A Short Introduction to TCP/IP and the Internet

Many excellent publications have been written on the topic of TCP/IP and the Internet. The aim of this appendix therefore is to provide only a short overview for the benefit of those readers who may not be familiar with the topic or who may desire a quick refresh.

We have included a selection of publications on advanced TCP/IP and Internet topics for your reference in C.14, "TCP/IP and Internet Publications" on page 245.

C.1 Why TCP/IP?

The need to interconnect networks that use different protocols was recognized early in the 1970s during a period when the use and development of networking technology was increasing. The rapid growth in networking over the past three decades has allowed users much greater access to resources and information as well as causing significant problems when merging, or interconnecting, different types of networks. Open protocols and common applications were required, leading to the development of a protocol suite known as *Transmission Control Protocol/Internet Protocol* (TCP/IP) which originated with the U.S. Department of Defense (DoD) in the mid-1960s and took its current form around 1978.

An interesting article about the history of the Internet can be found at the following URL:

`http://www.isoc.org/internet-history/`

C.2 The Growth of TCP/IP

In the early 1980s TCP/IP became the backbone protocol in multivendor networks such as ARPANET, NFSNET and regional networks. The protocol suite was integrated into the University of California at Berkeley's UNIX operating system and became available to the public for a nominal fee. From this point on TCP/IP has become widely used due to its inexpensive availability in UNIX and its spread to other operating systems, resulting in increasing use in both local area network (LAN) and wide area network (WAN) environments. Today, TCP/IP provides the ability for corporations to merge differing physical networks while giving users a common suite of functions. It allows interoperability between equipment supplied by multiple vendors on multiple platforms, and it provides access to the Internet.

In fact, the Internet, which has become the largest computer network in the world, is based on the TCP/IP protocol suite. The Internet consists of large international, national and regional backbone networks, that allow local and campus networks and individuals access to global resources. Use of the Internet has grown rapidly over the last few years, as illustrated in Table 7 on page 200. The most recent estimate has a number in excess of 29 million hosts on the Internet today.

Table 7. Internet Growth. The source of those figures can be found at the following URLs:

`http://www.isoc.org/guest/zakon/Internet/History/HIT.html#Growth`
`http://www.nw.com/zone/WWW/dist-bynum.html`

Date	Hosts	Networks	Domains
July 1989	130,000	650	3,900
July 1992	992,000	6,569	16,300
July 1993	1,776,000	13,767	26,000
July 1995	6,642,000	61,538	120,000
July 1996	12,881,000	134,365	488,000
July 1997	19,540,000	n/a	1,301,000

As opposed to the Internet, the term *intranet* has evolved recently to describe TCP/IP networks that are entirely under the control of a private authority or company. Those intranets may or may not have connections to other independent intranets (which would then be referred to as *extranets*) or the Internet. They may or may not be fully or partially visible to the outside depending on the implementation.

TCP/IP also provides for the routing of multiple protocols from and to diverse networks. For example, a requirement to connect isolated networks using IPX, AppleTalk and TCP/IP protocols using a single physical connection can be accomplished by using routers utilizing TCP/IP protocols.

One further reason for the growth of TCP/IP is the popularity of the socket programming interface, which is the programming interface between the TCP/IP transport protocol layer and TCP/IP applications. A large number of applications today have been written for the TCP/IP socket interface.

C.3 Internet Standards and Request for Comments (RFC)

We mentioned in the previous section that the Internet is a large multinational, multivendor, multiplatform network. That might give reason to ask some questions, such as:

- Are there any standards for such a diverse network?
- Who establishes and reviews them?
- Who assigns network addresses?
- Who manages the Internet?

The Internet Society (ISOC), formerly known as Internet Activities Board (IAB), is the non-profit, coordinating committee for Internet design, engineering and management. The ISOC members are committed to making the Internet function effectively and evolve to meet a large-scale, high-speed future. The ISOC holds several bodies for administering, standardizing, and researching for the Internet:

1. The Internet Architecture Board (IAB)
2. The Internet Engineering Task Force (IETF)
3. The Internet Research Task Force (IRTF)
4. The Internet Assigned Numbers Authority (IANA)

While the IAB oversees and manages the Request For Comments (RFC) publication process, the IETF actually defines the standards through a number of

subcommittees or task forces, and the IRTF engages in Internet-related research projects.

RFC is the mechanism through which the Internet protocol suite has been evolving. For example, an Internet protocol can have one of six states: standard, draft standard, proposed standard, experimental, informational and historic. In addition, an Internet protocol has one of five statuses: required, recommended, elective, limited use and not recommended. By communicating using the RFC, new protocols are being designed and implemented by researchers from both academic institutions and commercial corporations. At the same time, some old protocols are being superseded by new ones.

The RFC standards are described in the ″Internet Official Protocol Standards″ RFC, currently RFC 2200.

The task of coordinating the assignment of values to the parameters of protocols is delegated to the IANA. These protocol parameters include op-codes, type fields, terminal types, system names, object identifiers, and so on. The ″Assigned Numbers″ RFC, currently RFC 1700, documents these protocol parameters.

To obtain registered IP addresses (see C.5.1.1, “IP Addressing” on page 203) and domain names (see C.7.5, “Domain Name System (DNS)” on page 224), you need to contact the Internet Network Information Center (InterNIC), the administrative body for the Internet.

Registration is available online at the NIC Web site using the following URL:

`http://rs.internic.net/rs-internic.html`

C.4 TCP/IP Architecture

TCP/IP, as a set of communications protocols, is based on layers. Unlike SNA or OSI that distinguish seven layers of communication, there are only four layers in the TCP/IP model. They enable heterogeneous systems to communicate by performing network-related processing such as message routing, network control, error detection and correction.

The layering model of TCP/IP is shown in Figure 163 on page 202, with an explanation of each layer following thereafter:

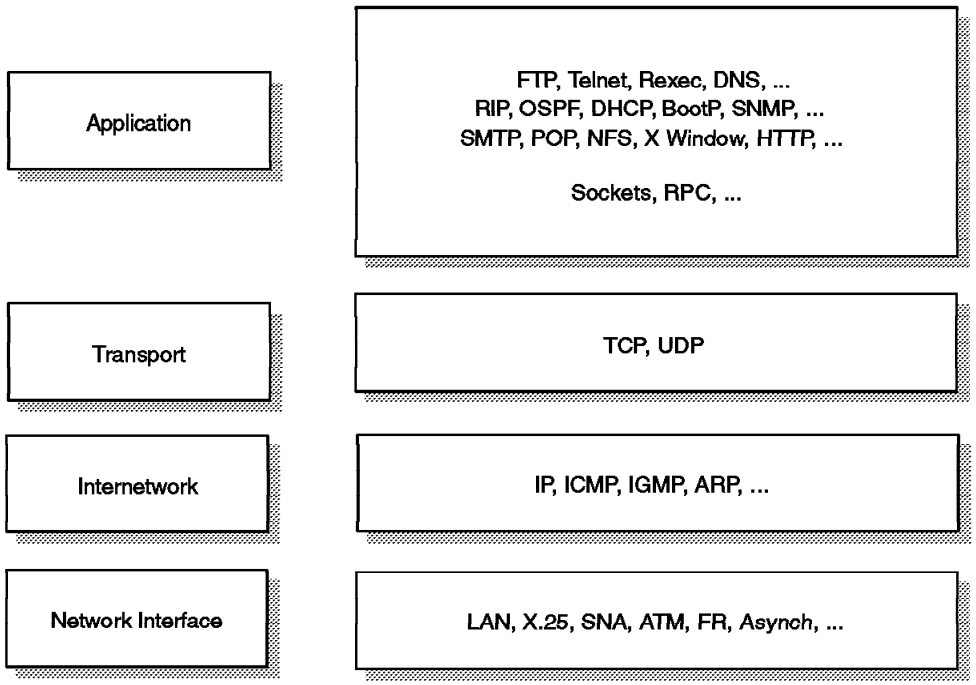


Figure 163. TCP/IP - Architecture Model: Layers and Protocols

Application Layer

The application layer is provided by the program that uses TCP/IP for communication. Examples of applications are Telnet, FTP, e-mail, Gopher and SMTP. The interface between the application and transport layers is defined by port numbers and sockets, which is described in more detail in C.6.1, “Ports and Sockets” on page 220.

Transport Layer

The transport layer provides communication between application programs. The applications may be on the same host or on different hosts. Multiple applications can be supported simultaneously. The transport layer is responsible for providing a reliable exchange of information. The main transport layer protocol is TCP. Another is User Datagram Protocol (UDP), which provides a connectionless service in comparison to TCP, which provides a connection-oriented service. That means that applications using UDP as the transport protocol have to provide their own end-to-end flow control. Usually, UDP is used by applications that need a fast transport mechanism.

Internet Layer

The Internet layer provides communication between computers. Part of communicating messages between computers is a routing function that ensures that messages will be correctly delivered to their destination. The Internet Protocol (IP) provides this routing function. Examples of Internet layer protocols are IP, ICMP, IGMP, ARP and RARP.

Network Interface Layer

The network interface layer, sometimes also referred to as link layer, data link layer or network layer, is implemented by the physical network that connects the computers. Examples are LAN (IEEE 802.x standards), Ethernet, X.25, ISDN, ATM, Frame Relay, or asynch.

Note that the RFCs actually do not describe or standardize any network layer protocols per se, they only standardize ways of accessing those protocols from the Internet layer.

C.5 TCP/IP Internet Layer Protocols

This section provides a short overview of the most important and common protocols of the TCP/IP Internet layer.

C.5.1 Internet Protocol (IP)

IP is the layer that hides the underlying physical network from the upper-layer protocols. It is an unreliable, best-effort and connectionless packet delivery protocol. Note that best-effort means that the packets sent by IP may be lost, out of order, or even duplicated, but IP will not handle these situations. It is up to the higher-layer protocols to deal with these situations.

One of the reasons for using a connectionless network protocol was to minimize the dependency on specific computing centers that used hierarchical connection-oriented networks. The DoD intended to deploy a network that would still be operational if parts of the country were destroyed. During earthquakes, this has been proved to be true for the Internet.

C.5.1.1 IP Addressing

IP uses *IP addresses* to specify source and target hosts on the Internet. (For example, we can contrast an IP address in TCP/IP with a fully qualified NETID.LUNAME in SNA) An IP address consists of 32 bits, which is usually represented in the form of four decimal numbers, one decimal number for each byte (or octet). For example:

```
00001001 01000011 00100110 00000001        a 32-bit address
    9       67       38       1             decimal notation (9.67.38.1)
```

An IP address consists of two logical parts: a network address and a host address. An IP address belongs to one of four classes depending on the value of its first four bits. (A fifth class, class E, is not commonly used.) This is shown in Figure 164 on page 204.

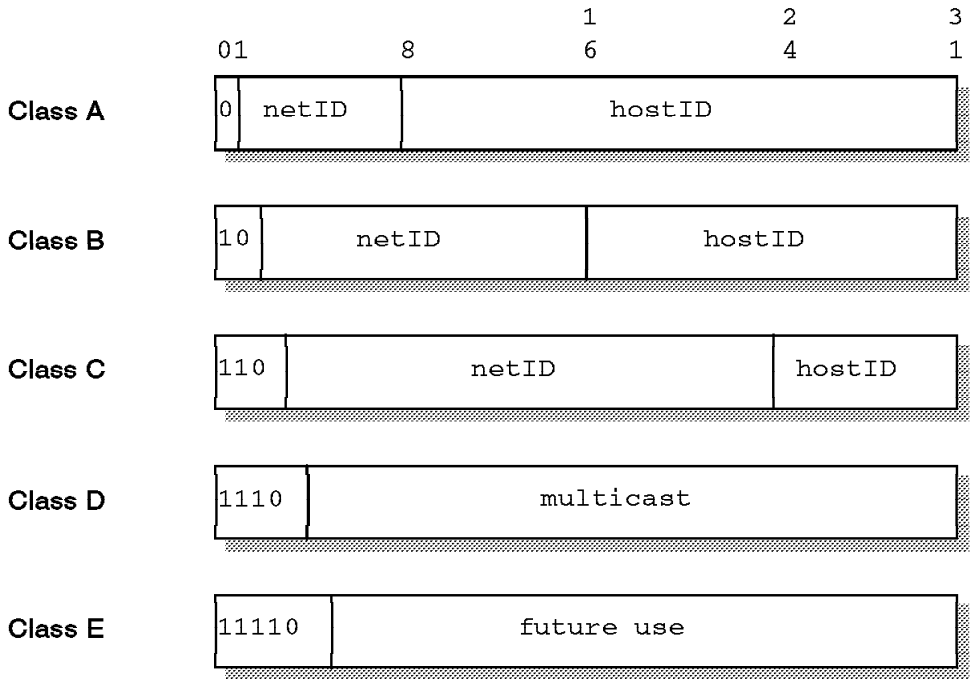


Figure 164. IP - Assigned Classes of IP Addresses

- Class A addresses use 7 bits for the <network> and 24 bits for the <host> portion of the IP address. That allows for 126 (2**7-2) networks with 16777214 (2**24-2) hosts each; a total of over 2 billion addresses.
- Class B addresses use 14 bits for the <network> and 16 bits for the <host> portion of the IP address. That allows for 16382 (2**14) networks with 65534 (2**16-2) hosts each; a total of over 1 billion addresses.
- Class C addresses use 21 bits for the <network> and 8 bits for the <host> portion of the IP address. That allows for 2097150 (2**21) networks with 254 (2**8-2) hosts each; a total of over half a billion addresses.
- Class D addresses are reserved for multicasting (a sort of broadcasting, but in a limited area, and only to hosts using the same class D address).
- Class E addresses are reserved for future use.

Some values for these host IDs and network IDs are pre-assigned and cannot be used for actual network or host addressing:

all bits 0 Stands for *this*: this host (IP address with <host address>=0) or this network (IP address with <network address>=0). When a host wants to communicate over a network, but does not yet know the network IP address, it may send packets with <network address>=0. Other hosts on the network will interpret the address as meaning *this network*. Their reply will contain the fully qualified network address, which the sender will record for future use.

all bits 1 stands for *all*: all networks or all hosts. For example:

128.2.255.255

means all hosts on network 128.2 (class B address).

This is called a directed broadcast address because it contains both a valid <network address> and a broadcast <host address>.

Loopback The class A network 127.0.0.0 is defined as the loopback network. Addresses from that network are assigned to interfaces which process data inside the local system and never access a physical network (loopback interfaces).

C.5.1.2 IP Subnets

Due to the explosive growth of the Internet, the principle of assigned IP addresses became too inflexible to allow easy changes to local network configurations. Those changes might occur when:

- A new type of physical network is installed at a location.
- Growth of the number of hosts requires splitting the local network into two or more separate networks.
- Growing distances require splitting a network into smaller networks, with gateways between them.

To avoid having to request additional IP network addresses in these cases, the concept of subnets was introduced. The assignment of subnets can be done locally, as the whole network still appears to be one IP network to the outside world.

Recall that an IP address consists of the pair <network address><host address>. For example, let us take a class A network; the address format is shown in Figure 165:

Figure 165. IP-Class A Address without Subnets

Let us use the following IP address:

```
00001001 01000011 00100110 00000001        a 32-bit address
   9       67       38       1             decimal notation (9.67.38.1)

          9.67.38.1          is an IP address  (class A)  having

          9                  as the <network address>
            67.38.1          as the <host address>
```

Subnets are an extension to this by considering a part of the <host address> to be a subnetwork address. IP addresses are then interpreted as <network address><subnetwork address><host address>.

We may, for example, wish to choose the bits from 8 to 25 of a class A IP address to indicate the subnet addresses, and the bits from 26 to 31 to indicate the actual host addresses. Figure 166 on page 206 shows the subnetted address that has thus derived from the original class A address:

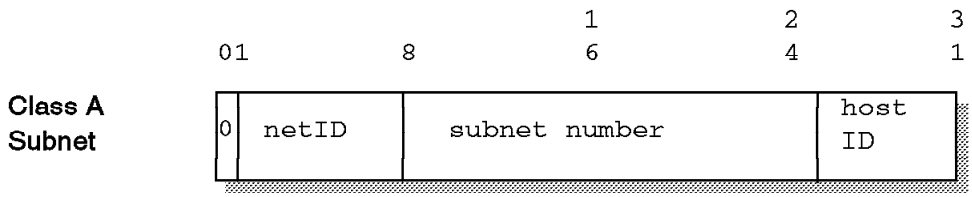


Figure 166. IP-Class A Address with Subnet Mask and Subnet Address

We normally use a bit mask, known as the subnet mask, to identify which bits of the original host address field to indicate the subnet number. In the above example, the subnet mask is 255.255.255.192 in decimal notation (or 11111111 11111111 11111111 11000000 in bit notation). Note that, by convention, the <network address> is masked as well.

For each of these subnet values, only (2**18)-2 addresses (from 1 to 262143) are valid because of the all bits 0 and all bits 1 number restrictions. This split will therefore give 262142 subnets each with a maximum of (2**6)-2 or 62 hosts.

You will notice that the value applied to the subnet number takes the value of the full byte with non-significant bits being set to zero. For example, the hexadecimal value 01 in this subnet mask assumes an 8-bit value 01000000 and gives a subnet value of 64 and not 1 as it might seem.

Applying this mask to our sample class A address 9.67.38.1 would break the address down as follows:

```
 00001001 01000011 00100110 00000001 = 9.67.38.1 (class A address)
 11111111 11111111 11111111 11------   255.255.255.192 (subnet mask)
==================================== logical_AND
 00001001 01000011 00100110 00------ = 9.67.38  (subnet base address)
```

and leaves a host address of:

```
 -------- -------- -------- --000001 = 1        (host address)
```

IP will recognize all host addresses as being on the local network for which the logical_AND operation described above produces the same result. This is important for routing IP datagrams in subnet environments (see C.5.1.4, “IP Routing” on page 208).

Note that the actual subnet number would be:

```
 -------- 01000011 00100110 00------ = 68760    (subnet number)
```

You will notice that the subnet number shown above is a relative number, that is, it is the 68760th subnet of network 9 with the given subnet mask. This number

bears no resemblance to the actual IP address that this host has been assigned (9.67.38.1) and has no meaning in terms of IP routing.

The division of the original <host address> part into <subnet> and <host> parts can be chosen freely by the local administrator; except that the values of all zeroes and all ones in the <subnet> field are reserved for special addresses.

Note: Because the range of available IP addresses is decreasing rapidly, many routers do support the use of all zeroes and all ones in the <subnet> field, though this is not coherent with the standards.

C.5.1.3 IP Datagram

The unit of transfer of a data packet in TCP/IP is called an IP datagram. It is made up of a header containing information for IP and data that is only relevant to the higher level protocols. IP can handle fragmentation and re-assembly of IP datagrams. The maximum length of an IP datagram is 65,535 bytes (or octets). There is also a requirement for all TCP/IP hosts to support IP datagrams of size up to 576 bytes without fragmentation.

The IP datagram header is a minimum of 20 bytes long:

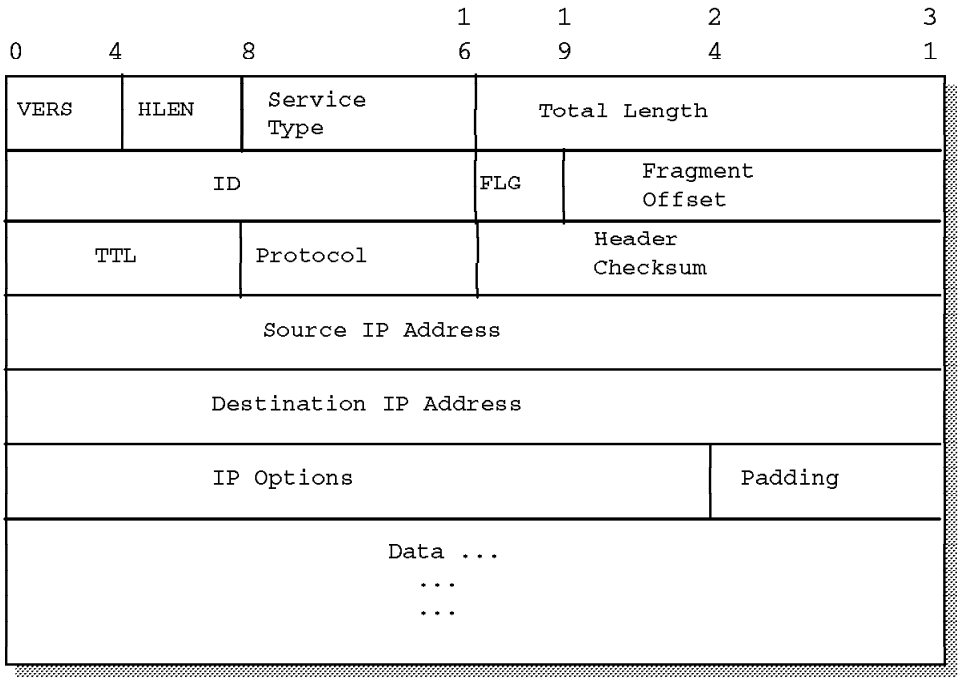


Figure 167. IP - Format of an IP Datagram Header

We do not elaborate on the format of the IP datagram header. You can find this information in the listed publications.

C.5.1.4 IP Routing

There are two types of IP routing: direct and indirect.

Direct Routing: If the destination host is attached to a physical network to which the source host is also attached, an IP datagram can be sent directly, simply by encapsulating the IP datagram in the physical network frame. This is called direct delivery and is referred to as direct routing.

Indirect Routing: Indirect routing occurs when the destination host is not on a network directly attached to the source host. The only way to reach the destination is via one or more IP gateways (note that in TCP/IP terminology, the terms gateway and router are used interchangibly for a system that actually performs the duties of a router). The address of the first of these gateways (the first hop) is called an indirect route in the context of the IP routing algorithm. The address of the first gateway is the only information needed by the source host.

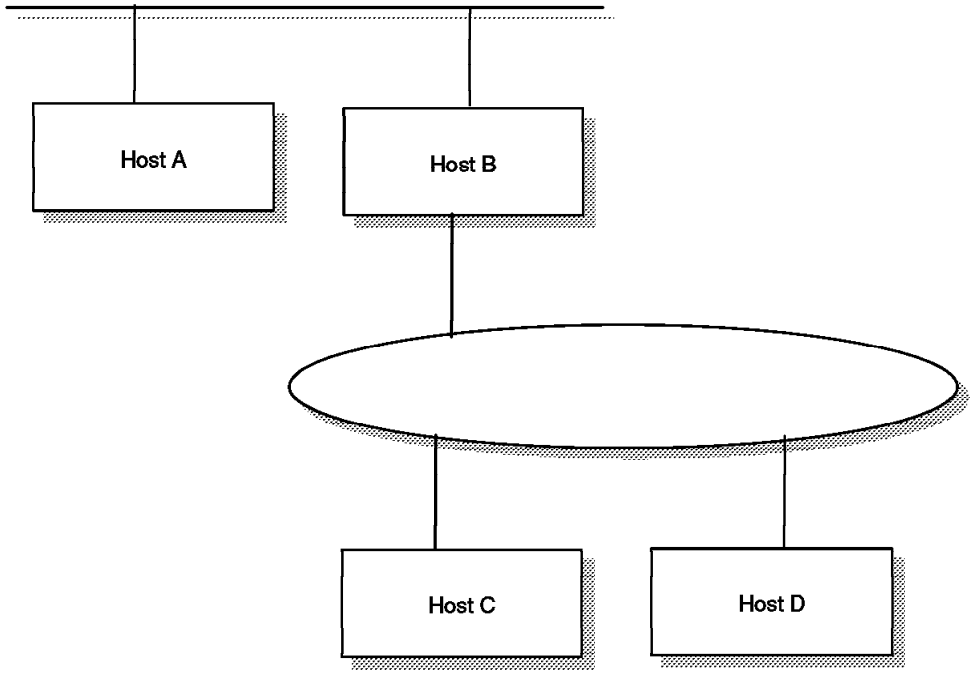


Figure 168. IP - Direct and Indirect Routes. (Host C has a direct route to hosts B and D, and an indirect route to host A via gateway B).

IP Routing Table: The determination of available direct routes is derived from the list of local interfaces available to IP and is composed by IP automatically at initialization. A list of networks and associated gateways (indirect routes) needs to be configured to be used with IP routing if required. Each host keeps the set of mappings between the following:

- Destination IP network address(es)
- Route(s) to next gateway(s)

These are stored in a table called the IP routing table. Three types of mappings can be found in this table:

1. The direct routes, for locally attached networks.
2. The indirect routes, for networks reachable via one or more gateways.
3. The default route, which contains the (direct or indirect) route to be used in case the destination IP network is not found in the mappings of type 1 and 2 above.

See the network in Figure 169 for an example configuration.

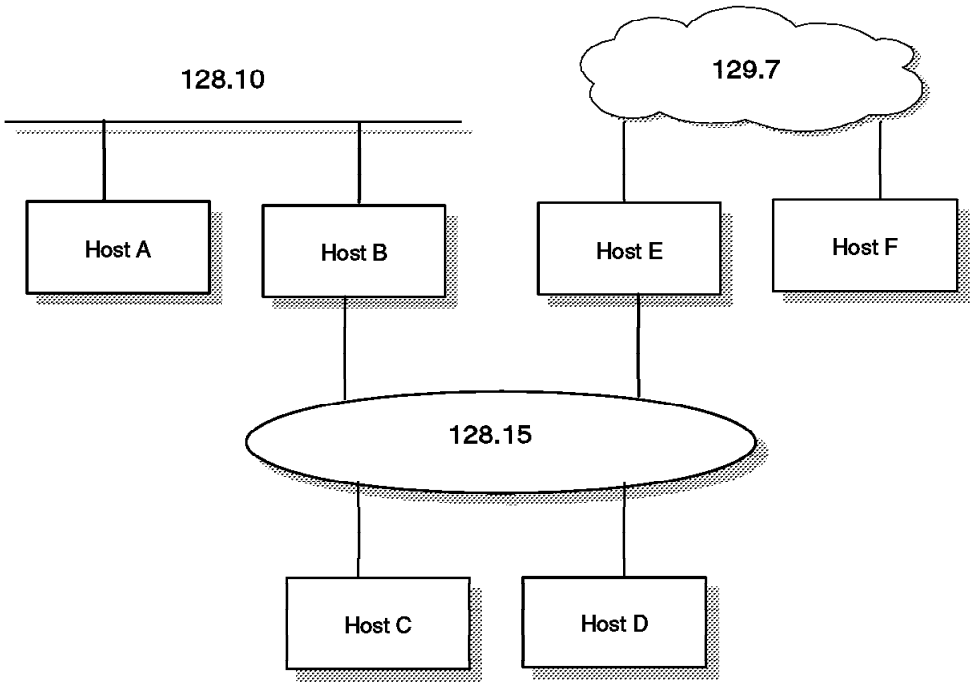


Figure 169. IP - Routing Table Scenario

The routing table of host D might contain the following (symbolic) entries:

```
destination      router    interface

    129.7.0.0    E         lan0
    128.15.0.0   D         lan0
    128.10.0.0   B         lan0
    default      B         lan0
    127.0.0.1    loopback  lo
```

Figure 170. IP - Routing Table Example 1

The routing table of host F might contain the following (symbolic) entries:

```
destination      router    interface

    129.7.0.0    F         wan0
    default      E         wan0
    127.0.0.1    loopback  lo
```

Figure 171. IP - Routing Table Example 2

IP Routing Algorithm: IP uses a unique algorithm to route an IP datagram. It is called the IP routing algorithm which is illustrated in the figure below, including support of subnets:

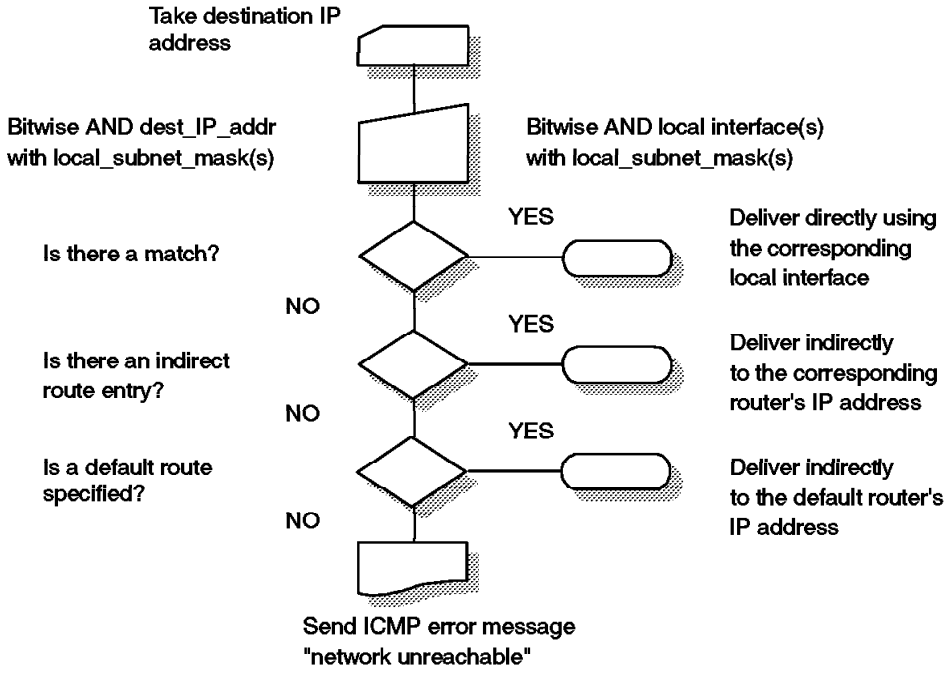


Figure 172. IP - Routing Algorithm (with Subnets)

Notes:

1. This is an iterative process. It is applied by every host handling a datagram, except for the host to which the datagram is finally delivered.
2. Routing tables and the routing algorithm are local to any host in an IP network. In order to be able to forward IP datagrams on behalf of other hosts, routers need to exchange their routing table information with other routers in the network. This is done using special routing protocols, some of which are discussed in C.9, “TCP/IP Routing Protocols and Techniques” on page 232.

C.5.2 Internet Control Message Protocol (ICMP)

Although ICMP is shown in Figure 164 on page 204 as being in the same protocol layer as IP, it is actually an integral part of IP. ICMP is used for reporting errors in datagram delivery, such as destination unreachable, and it can assist in discovering routers and maximum transmission units (MTU) along a path that an IP datagram eventually travels.

Perhaps one of the most useful commands available on all TCP/IP implementations is the PING (Packet INternet Groper) application. PING uses ICMP to send an Echo datagram to a specified IP address and wait for it to return. This is very useful for debugging purposes and also for knowing if a remote host can be reached from the local host. ICMP is defined in RFC 792.

C.5.3 Internet Group Management Protocol (IGMP) and IP Multicasting

Similar to ICMP, the Internet Group Management Protocol (IGMP) is also an integral part of IP. It serves the purpose of allowing hosts to participate in IP multicasts and to cancel such participation. IGMP further provides routers with the capability to check if any hosts on a local subnet are at all interested in a particular multicast.

As opposed to broadcasting, IP multicasting provides a way to spread information across a network assuring that only those will receive it that are interested in that information. Though related data will potentially travel through the whole network, all hosts or even whole subnets that are not interested in a particular multicast will not receive any related messages. This method significantly reduces network traffic in two ways:

1. Servers only have to send information once without requiring to know who will finally receive it.
2. Hosts, or whole subnets, that are not interested, do not receive packages to process as would be the case with broadcasts.

In order to receive multicast messages, a host must join a multicast group by assigning a specific class D IP address to one or more of its interfaces on behalf of the application that wants to receive the messages. See Figure 164 on page 204 for an illustration of class D IP addresses that are reserved solely for the purpose of multicasting. Hosts then announce to adjacent routers the multicast groups they have joined (or, to say it in a more modern way, what channels they are interested in).

Routers form a spanning tree starting from the network where the multicast server is located. They keep track of active channels on their local subnet(s) and forward any related traffic as long as there is at least one host that is interested. Routers periodically check if there are any active channels on their local subnet(s), and if not, they prune themselves out of the multicast spanning tree.

Note: A single router may be part of many such spanning trees and can prune trees independent from one another.

Routers among each other use special purpose multicast routing protocols, such as Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), or Protocol Independent Multicast (PIM), to form the spanning trees and to decide where and if to deliver multicast packets.

See RFC 966, RFC 1112 and RFC 1458 for more information about IP multicasting.

C.5.4 Interfacing with the Network Layer

Though the network interface layer itself is not covered by the TCP/IP standards, the RFCs do specify certain methods to access that layer from the higher layers. Before we describe some of the protocols that interface with the network layer, we need to distinguish between different types of networks that the Internet layer can be connected with:

Multiaccess broadcast networks

In a network of this type, any system (TCP/IP host) can have multiple connections to other hosts simultaneously, and it can also send information to all other hosts on the same network with a single, special kind of message (broadcast). Local area networks (LANs) typically represent this type of network. Protocols such as ARP, ProxyARP, RARP, BootP and DHCP are used with this type of network. We will briefly describe some of them in this and following sections.

Multiaccess non-broadcast networks

In a network of this type, any host can have multiple connections to other hosts simultaneously but there are no broadcast mechanisms in place. Examples of this type of network are X.25, Frame Relay and AnyNet Sockets over SNA.

Point-to-point networks

In a network of this type, a host can only have one connection to one other host at any time, and there are no broadcast mechanisms in place. Examples of this type of network are SNAlink and asynchronous connections (using SLIP or PPP which are briefly described in this section).

Notes:

1. The term *connection* in the three paragraphs before applies to any single IP interface of a host in any of the network types mentioned. For instance, a host could have multiple point-to-point interfaces and thus more than one connection at a time, but still only one per interface.
2. Some publications only distinguish between broadcast and non-broadcast networks.

C.5.4.1 Hardware Address Resolution (ARP and RARP)

The Address Resolution Protocol (ARP) maps Internet addresses to hardware addresses. When an application attempts to send data over a TCP/IP network capable of broadcasting, IP requests the appropriate hardware address mapping using ARP. If the mapping is not in the mapping table (ARP cache), an ARP broadcast packet is sent to all the hosts on the network requesting the physical hardware address for the host. For more information about ARP, see RFC 826.

An exception to the rule constitutes the Asynchronous Transfer Mode (ATM) technology where ARP cannot be implemented in the physical layer as described above. Therefore, an ARP server is used with which every host has to register upon initialization in order to be able to resolve IP addresses to hardware addresses.

Some network hosts do not know their IP addresses when they are initialized. This can especially be true in the case of a host needing to be booted from diskette. Reverse ARP (RARP) can be used by, for example, a diskless workstation to determine its own IP address. In this case the workstation would already know its hardware address (discovered at initialization) and would

broadcast a request to a RARP server to map the addresses. It is necessary to have a RARP server in your network in order to implement RARP.

C.5.4.2 Serial Line Interface Protocol (SLIP)

The Serial Line Internet Protocol (SLIP) allows you to set up a point-to-point connection between two TCP/IP hosts over a serial line, for example, a serial cable or a RS-232 connection into a modem and over a telephone line. You can use SLIP to access a remote TCP/IP network (such as a service provider′s network) from your local host or to route datagrams between two TCP/IP networks. For more information about SLIP, see RFC 1055.

SLIP has several deficiencies, such as:

- Only being able to transport IP datagrams. (It can not be used to route other protocols.)
- Having no ability to determine the address of the host at the other end of the connection. (Both hosts must know each other′s addresses.)
- Having no error correction or data compression facility, therefore being unreliable across noisy and low speed lines.

SLIP is not an Internet standard, but it is in widespread use. Due to the fact that most implementations incorporate SLIP and that for many applications the issues listed above are not important, it is a good choice for remote connection (or dial-in access). However, for use between hosts across a dynamic environment such as a large WAN, the problems are a major consideration and make SLIP an inadequate protocol to link routers.

C.5.4.3 Point-to-Point Protocol (PPP)

The Point-to-Point Protocol (PPP) is an Internet standard that has been developed to overcome the problems associated with SLIP. PPP allows addresses to be negotiated across a connection instead of being statically defined; this is not broadcasting because the negotiation is limited to a single link rather than to all hosts.

PPP implements reliable delivery of datagrams over both synchronous and asynchronous serial lines. It also allows compression to be negotiated and can be used to route a wide variety of network protocols. Another feature of PPP, called multilink PPP, allows for the combination of several point-to-point connections between the same hosts to appear as one logical connection in order to increase available network bandwidth. For more information about PPP, see RFCs 1717 and 1661.

Note: Do not confuse the terms point-to-point connection and Point-to-Point Protocol (PPP). A point-to-point connection is a link between two specific interfaces, while PPP is a protocol used to communicate over the link.

C.5.5 The Future Version of IP (IPv6)

It has been mentioned in C.5.1.1, “IP Addressing” on page 203 that the address space of the current version of IP, IP Version 4 or IPv4 allows for almost four billions of valid addresses. One might think that this should be sufficient to cope with the growth of many more years. The truth is that, due to the impacts of growth as well as the restrictions of subnetting, the IP address space is nearing exhaustion between the years 2005 and 2011.

Apart from the problem of running out of IP addresses, there are other restrictions that called for a definition of a new IP protocol:

1. Even with the use of CIDR (see C.9.3, “Classless Inter-Domain Routing (CIDR)” on page 233), routing tables, primarily in the IP backbone routers, are growing too large to be manageable.
2. Traffic priority, or class of service, is vaguely defined, scarcely used and not at all enforced in IPv4 but highly desirable for modern real-time applications. (See C.12, “Real-Time and Multimedia Application Support” on page 243.)
3. Even with BootP and DHCP, IP address administration remains a time-consuming task. (See C.8.2, “Dynamic Host Configuration Protocol (DHCP)” on page 230.)

The IETF therefore initiated the IPng (IP next generation) working group that finally came up with a protocol specification, IP Version 6 (IPv6), that offers the following benefits:

- A dramatically larger address space, said to be sufficient for the next 30 years.
- Globally unique and hierarchical addressing, based on prefixes rather than on address classes to keep routing tables small and backbone routing efficient.
- Multicasting instead of broadcasting.
- Class of service to distinguish between different types of traffic.
- A built-in mechanism for autoconfiguration of network interfaces.
- Built-in authentication and encryption.
- Encapsulation of itself and other protocols.
- Transition methods to migrate from IPv4.
- Compatibility methods to coexist and communicate with IPv4.

The specifications of IPv6 and associated protocols and issues can be found in RFCs 1883 to 1887.

C.5.5.1 IPv6 Addressing

IPv6 uses 128-bit addresses instead of the 32-bit addresses of IPv4. That theoretically allows for as many as 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses. Even when used with the same efficiency as today′s IPv4 address space, that would still allow for 50,000 addresses per square meter of land on Earth.

IPv6 addresses are represented in the form of eight hexadecimal numbers divided by colons, for example:

```
FE80:0000:0000:0000:0001:0800:23e7:f5db
```

To shorten the notation of addresses, leading zeroes in any of the groups can be omitted, for example:

```
FE80:0:0:0:1:800:23e7:f5db
```

Finally, a group of all zeroes, or consecutive groups of all zeroes, can be substituted by a double colon, for example:

```
FE80::1:800:23e7:f5db
```

Note: The double colon shortcut may be used only once in the notation of an IPv6 address. If there are more groups of all zeroes that are not

consecutive, only one may be substituted by the double colon, the others would have to be noted as 0.

The IPv6 address space is organized using format prefixes, similar to telephone country and area codes, that logically divide it in the form of a tree so that a route from one network to another can easily be found. The following prefixes have been assigned so far:

Table 8. IPv6 - Format Prefix Allocation

Allocation	Prefix (bin)	Prefix (hex)	Mask Length (bits)	Fraction of Address Space
Reserved	0000 0000	0:: /8	8	1/256
Reserved for NSAP	0000 001	200:: /7	7	1/128
Reserved for IPX	0000 010	400:: /7	7	1/128
Provider-based Unicast	010	4000:: /3	3	1/8
Geography-based Unicast	100	8000:: /3	3	1/8
Link-local Unicast	1111 1110 10	FE80:: /10	10	1/1024
Site-local Unicast	1111 1110 11	FEC0:: /10	10	1/1024
Multicast	1111 1111	FF00:: /8	8	1/256
Total Allocation				about 28%

IPv6 defines the following types of addresses:

Unicast Address

A unicast address is an identifier assigned to a single interface. Packets send to that address will only be delivered to that interface. Special purpose unicast addresses are defined as follows:

Loopback address (::1) This address is assigned to a virtual interface over which a host can send packets only to itself. It is equivalent to the IPv4 loopback address 127.0.0.1.

Unspecified address (::) This address is used as a source address by hosts while performing autoconfiguration. It is equivalent to the IPv4 unspecified address 0.0.0.0.

IPv4-embedded address (::<IPv4_address>) Addresses of this kind are used when IPv6 traffic needs to be tunneled across existing IPv4 networks. The endpoint of such tunnels can be either hosts (automatic tunneling) or routers (configured tunneling).

IPv4-mapped address (::FFFF:<IPv4_address>) Addresses of this kind are used when an IPv6 host needs to communicate with an IPv4 host. This requires a dual stack host or router for header translations.

Link-local address Addresses of this kind can be used only on the physical network that a host's interface is attached to.

Site-local address Addresses of this kind cannot be routed into the Internet. They are the equivalent of IPv4 networks for private use (10.0.0.0, 176.16.0.0-176.31.0.0, 192.168.0.0-192.168.255.0).

Provider-based unicast addresses are administered by Internet Service Providers (ISPs), whereas geography-based unicast addresses are administered by the IANA or other global registries. Both address types can be used on the Internet and are equivalent to IPv4 addresses that have been registered with the InterNIC. Figure 173 illustrates the format of a provider-based unicast address as defined by RFC 2073:

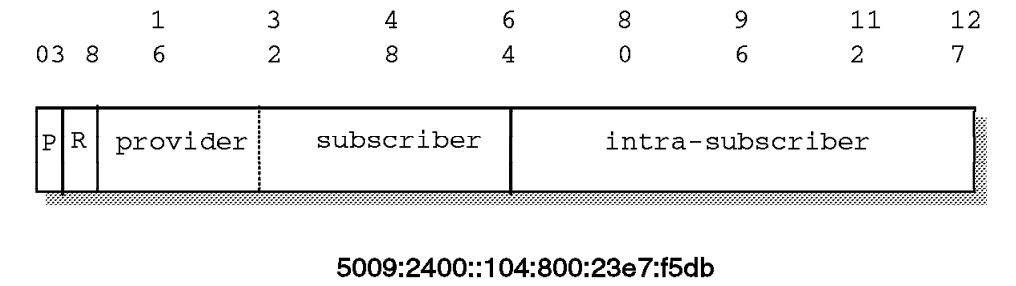


Figure 173. IPv6 - Provider-Based Unicast Address Format

Multicast Address

A multicast address is an identifier assigned to a set of interfaces on multiple host. Packets sent to that address will be delivered to all interfaces with that address. (See C.5.3, “Internet Group Management Protocol (IGMP) and IP Multicasting” on page 211 for more information on IP multicasting.) IPv6 defines two kinds of multicast addresses:

Permanent Assigned by the IANA and reserved for special purposes.
Transient Addresses of this kind can be established by applications, such as video multicasting, as required. When the application ends, the address will be released by the application and can be reused.

Special purpose multicast addresses are defined as follows:

All systems node-local (FF01::1) Defines all systems to the host itself.
All systems link-local (FF02::1) Defines all systems on the local network.
All routers node-local (FF01::2) Defines all routers to the host itself.
All routers link-local (FF02::2) Defines all routers on the local network.
All DHCP servers and relay agents (FF02::1:0) Defines all DHCP servers and BootP relay agents on the local network.
Solicited node address (FF02::1 /96) Defines a special multicast address that is derived from the last 32 bits of an IPv6 address and the shown prefix.

Anycast Address

An anycast address is a special type of unicast address that can be assigned to interfaces on multiple hosts. Packets sent so such an address will be delivered to the nearest interface with that address. Routers determine the nearest interface based upon their definition of distance, for example hops in case of RIP or link state in case of OSPF.

C.5.5.2 IPv6 Datagram

To further speed up routing in IPv6, the format of the datagram header has been simplified from its counterpart in IPv4 (shown in Figure 167 on page 207). This significantly reduces the processing time per datagram in a router because less fields and flags have to be looked at and acted upon, no header checksums have to be calculated, and most of the traffic passing through routers is unicast traffic without special options anyway. The format of the IPv6 datagram header is shown in Figure 174.

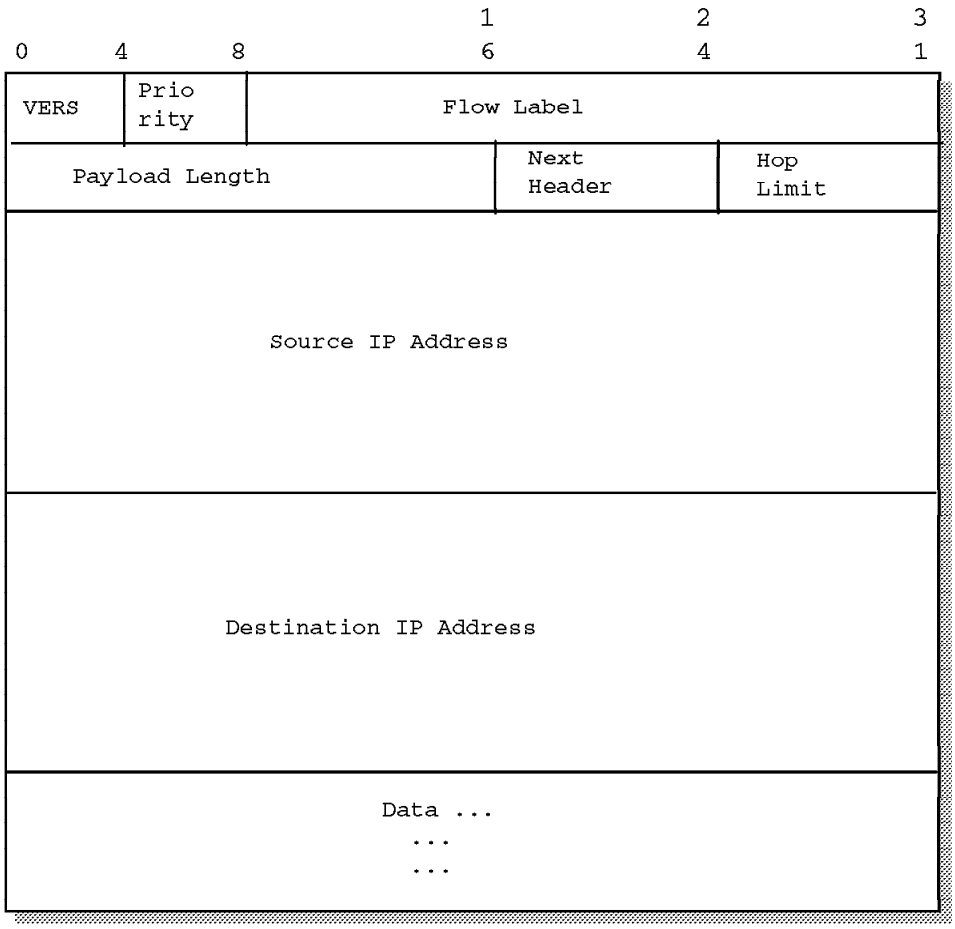


Figure 174. IPv6 - Format of an IPv6 Datagram Header

Simplifying the header does not mean that IPv6 has less functionality then IPv4. Special purpose fields and options, such as fragmentation or source routing, have been taken out of the basic datagram header. They can be appended as extension headers whenever required. Such extension headers may also carry information about authentication and encryption.

IPv6 introduces the concept of class of service which allows applications to specify a certain priority for the traffic they generate. IPv4-based routers normally treat all traffic equal, whereas IPv6-based routers now must act on such prioritized packets in the following way:

1. For low priorities 0 to 7, start dropping packets when the network becomes congested (congestion-controlled).
2. For high priorities 8 to 15, try to forward packets even when the network is becoming congested by dropping packets with lower priority (non congestion-controlled). Real-time applications would opt for this range of priority.

New to the IPv6 header is the identification of flows. A flow specifies a subsequent number of packets that have the same label (chosen at random by the sending host), destination address and routing options. That allows routers to cache information about how to route such packets in special table in memory that is easier to look up than its routing tables. Flows can also be significant for traffic that relies on high speed, such as real-time applications. Flows and priority together could also be used for resource reservation, similar to RSVP (see C.12.1, “Resource Reservation Protocol (RSVP)” on page 243).

Also defined with IPv6 and placed in extension headers are authentication and encryption methods according to the IP Security Architecture (IPSec). Please see C.11.3, “IP Security Architecture (IPSec)” on page 241 for a more detailed descriptions on IPSec.

C.5.5.3 Internet Control Message Protocol Version 6 (ICMPv6)

Similar to ICMP and IP, ICMPv6 is a corresponding message protocol which is an integral part of IPv6. ICMPv6 combines three former IPv4 protocols:

1. ICMP
2. IGMP
3. ARP

Therefore, ICMPv6 is not only used for conveying error messages and echos (PING), but also to resolve IP to hardware address mappings (ARP) and to control multicast group memberships (IGMP).

Since there are no broadcasts in IPv6 anymore, ICMPv6 uses the solicited node multicast address so that a host who wants to communicate with another host on the same network can obtain that other host′s hardware address. This process is called neighbor solicitation and is equivalent to ARP in IPv4. Please see C.5.4.1, “Hardware Address Resolution (ARP and RARP)” on page 212 and RFC 1970 for more information.

C.5.5.4 IPv6 Autoconfiguration

New with IPv6 is the capability to automatically assign an address (a link-local unicast address) to an interface at initialization time. The idea behind this mechanism is to provide a network that can become operational with minimal to no administrator action. A host configured in that way should be able to communicate in its local physical network(s) and may eventually learn enough information to communicate across routers, and even across the Internet.

The following configuration processes are defined:

Stateless Autoconfiguration

This process is described in RFC 1971 and defines what should happen when a host initializes its interfaces with IPv6:

1. Obtain an interface token from the interface hardware, for instance a 48-bit MAC address on token-ring or Ethernet networks.

2. Concatenate the link-local unicast prefix (FE80:: /10) with that interface token to form a link-local unicast address.
3. Perform a duplicate address check by pinging the solicited node multicast address that has been derived from the previously created address. It that test is passed successfully, the host can actively use that IP address.
4. Listen for neighbor notification and router advertisement messages to obtain additional network prefixes and information on how and if to proceed with the autoconfiguration process.
5. Concatenate any received network prefixes with the interface token to form additional interface addresses.
6. Establish the loopback interface.
7. Join all default multicast groups.

Stateful Autoconfiguration

This process defines what a host can or should do in addition to stateless autoconfiguration, or when the stateless process does not work or should not be used. Router advertisements or an administrators configuration instruct hosts if they should engage in the stateful process:

- Use DHCP to obtain IP address and other relevant configuration parameters.
- Use stateless autoconfiguration to obtain an interface address and then use DHCP to get additional configuration parameters.
- Use DHCP for everything because no interface token could be obtained from the interface hardware.
- Use DHCP for everything because the duplicate address check has failed.

Router Advertisement and Discovery

IPv6 routers periodically announce their presence by sending router advertisement messages. A host listens for such messages during stateless autoconfiguration to learn any network prefixes, link MTU sizes, and how to proceed with its configuration process. If a host does not receive router advertisement messages, it can request information about available routers on the physical network by sending a router solicitation message to the all routers link-local multicast address.

Manual Configuration

If all else fails, an address must be manually assigned to an interface.

IPv6 is just facing its initial implementations in commercially available hardware and software products, but there are no IPv6 networks available to the general public yet. Internet providers are, however, expected to open IPv6 networks in the 1998/1999 timeframe.

More information about IPv6 networks can be found at the following URLs:

`http://playground.sun.com/ipng/`
`http://www-6bone.lbl.gov/6bone/`

C.6 TCP/IP Transport Layer Protocols and Interfaces

This section provides a brief overview of the protocols of the TCP/IP transport layer.

C.6.1 Ports and Sockets

Each process that wants to communicate with another process identifies itself to the TCP/IP protocol suite by one or more ports. A port is a 16-bit number, used by the host-to-host protocol to identify to which higher-level protocol or application program (process) it must deliver incoming messages.

As some higher-level programs are themselves protocols, standardized in the TCP/IP protocol suite, such as Telnet and FTP, they use the same port number in all TCP/IP implementations. (Port 23 is used by a Telnet server; ports 20 and 21 are used by an FTP server.) Those assigned port numbers are called well-known ports and the standard applications are called well-known services.

The well-known ports are controlled and assigned by the IANA and on most systems can only be used by system processes or by programs executed by privileged users. The assigned well-known ports occupy port numbers in the range 0 to 1023. The ports with numbers in the range 1024-65535 are not controlled by the Internet central authority and on most systems can be used by ordinary user-developed programs.

Confusion due to two different applications trying to use the same port numbers on one host is avoided by writing those applications to request an available port from TCP/IP. Because this port number is dynamically assigned, it may differ from one invocation of an application to the next.

UDP, TCP and ISO TP-4 all use the same port principle. To the extent possible, the same port numbers are used for the same services on top of UDP, TCP and ISO TP-4.

A socket is a special type of file handle that is used by a process to request network services from the operating system. A socket address (also known as transport address or half association) is the triple {protocol, local_address, local_port_number}, or {protocol, foreign_address, foreign_port_number}. In the TCP/IP suite, for example, {tcp, 193.44.234.3, 12345}. That is, a socket is an end point for communication that can be named and addressed in a network.

C.6.2 The Sockets Application Programming Interface

The socket interface is one of several application programming interfaces (APIs) to the communication protocols. Designed to be a generic communication programming interface, it was first introduced by the 4.2BSD UNIX system. Although it has not been standardized, it has become a de facto industry standard.

The socket interface is differentiated by the services that are provided to applications: stream sockets (connection-oriented), datagram sockets (connectionless), and raw sockets (direct access to lower-layer protocols) services.

A variation of the BSD sockets interface is provided by the Winsock interface developed by Microsoft and other vendors to support TCP/IP applications on

Windows operating systems. Winsock 2.0, the latest version, provides a more generalized interface allowing applications to communicate with any available transport layer protocol and underlying network services, including, but no longer limited to, TCP/IP.

C.6.3 User Datagram Protocol (UDP)

UDP is basically an application interface to IP. It provides no additional reliability, flow-control or error recovery. It simply serves as a multiplexer/demultiplexer for sending/receiving IP datagrams, using ports to direct the datagrams. As a result of this, the maximum amount of data that can be transferred by UDP at any time is what fits within a single IP datagram, less the size of the UDP header. This typically makes UDP a good choice for messaging and querying applications that do not rely on immediate acknowledgements, such as SMTP, DNS and SNMP. (Though such applications might also use TCP if so implemented.)

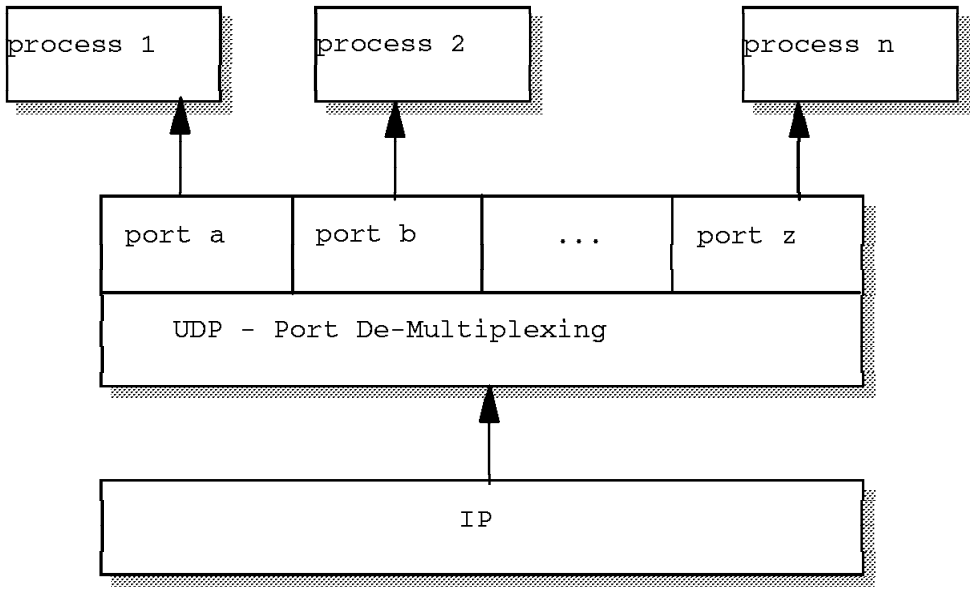


Figure 175. UDP - Demultiplexing Based on Ports

C.6.4 Transmission Control Protocol (TCP)

We have previously discussed IP, the unreliable connectionless packet (datagram) delivery mechanism that forms the basis of the TCP/IP protocol suite. TCP is the higher-level protocol that provides reliability, flow control and some error recovery. Many of the TCP/IP application protocols, such as Telnet and FTP, use TCP as the underlying protocol.

TCP is a connection-oriented, end-to-end reliable protocol providing logical connections between pairs of processes. Within TCP, a connection is uniquely defined by a pair of sockets (that is, by a pair of processes, on the same or different systems, that are exchanging information).

The two processes communicate with each other over a TCP connection (InterProcess Communication - IPC), as shown in Figure 176 on page 222.

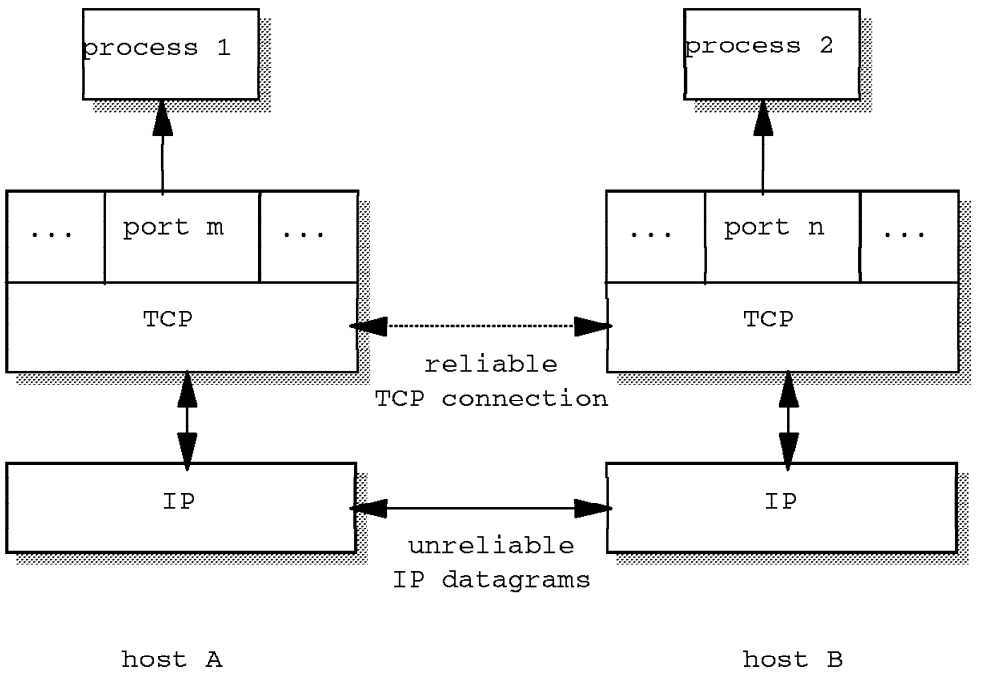


Figure 176. TCP-Connection Between Processes. (Processes 1 and 2 communicate over a TCP connection carried by IP datagrams.)

TCP does not recognize any application's data patterns but treats data as a stream of bytes that the sending application writes to a buffer known as the TCP window. The following steps explain TCP operation in a simplified way:

1. During connection establishment, the two hosts agree on an initial window size for that connection. TCP on the sending system then fills up the window with application data, chops its content into packets that conveniently fit into IP datagrams and passes those on to IP to be submitted to the receiver.
2. TCP on the receiving system puts the packets back in order, sends acknowledges for every two packets received in order and fills up a receive buffer (window) from where the receiving application takes out the incoming data stream.
3. TCP on the sending system advances the window for as many packets as have been acknowledged (in order) and continues to transmit packages until the end of the data stream is reached. If packets are not acknowledged, TCP will wait and retransmit them before notifying the application of a communications problem. While TCP awaits acknowledgements, the window cannot be advanced and packet transfer will slow down.
4. TCP provides flow control during an ongoing communication by varying send and receive window sizes at each transfer as required, which is important if a system runs out of memory for buffer space, or when a very fast system is communicating with a very slow system.

5. TCP also provides for urgent data (interrupt signalling) to travel ahead of communication traffic on the current connection.
6. When the sending application terminates, normally or abnormally, TCP will try to gracefully shut down the connection.

C.7 TCP/IP Application Protocols

One of the reasons why TCP/IP is so popular is that there are many simple and useful standard applications available. We summarize several common TCP/IP applications in this section.

C.7.1 Remote Login and Terminal Emulation (Telnet)

Telnet (teletypewriter network) is the virtual terminal protocol in TCP/IP. It allows users of one host to log into a remote host and interact as normal terminal users of that host. Telnet is a line-oriented protocol using the ASCII character set. Though initially designed for terminal emulation, Telnet is also used as the underlying protocol for file transfer (FTP control sessions) and e-mail (SMTP) operations.

For readers who are familiar with SNA, we can relate Telnet in TCP/IP to the terminal emulators (3270 or 5250 types) in SNA. In fact, all of the IBM TCP/IP product implementations provide Telnet support of 3270 terminal emulation in addition to the many other terminal emulation protocols, such as the widely used DEC VT terminal emulation types.

3270 terminal emulation differs from normal Telnet operation in the following ways:

1. It uses block-mode rather than line-mode.
2. It uses EBCDIC character set rather than ASCII character set.
3. It uses special key functions such as ATTN and SYSREQ.

A 3270 Telnet (TN3270) server must support those characteristics during initial client/server session negotiations. TN3270 sessions can represent either display or printer devices.

Originally TN3270 sessions were identified as non-SNA devices to a mainframe computer. The TN3270E extensions (see RFC 1647) define methods to map TN3270 sessions to specific SNA logical unit (LU) names thus effectively turning them into SNA devices. This eases the use of certain applications and allows users to be assigned the same LUs whenever they connect to the server.

C.7.2 File Transfer Protocols (FTP and TFTP)

FTP (File Transfer Protocol) provides the functions of transferring files between two TCP/IP hosts. Since FTP is built on the services of TCP in the transport layer, it provides a reliable and end-to-end connection during the file transfer operation. Security is provided by the normal user ID and password authentication.

TFTP (Trivial File Transfer Protocol) is a somewhat simplified companion of FTP. It operates on UDP and therefore does not guarantee reliable end-to-end connection or delivery. It also offers only limited security based on client hostname authorization. Nonetheless, TFTP is quite commonly used in conjunction with BOOTP to distribute startup program code to diskless network

stations (see C.8.1, “Bootstrap Protocol (BOOTP)” on page 229 for more information on BOOTP).

C.7.3 Remote Printing (LPR and LPD)

The line printer requester (LPR) allows access to printers on other computers running the line printer daemon (LPD) as though they were on your computer. The clients provided (LPR, LPQ, LPRM or LPRMON or LPRPORTD) allow the user to send files or redirect printer output to a remote host running a remote print server (LPD). Some of these clients can also be used to query the status of a job, as well as to delegate a job. For more information about remote printing, see RFC 1179.

C.7.4 Remote Command Execution (REXEC and RSH)

Remote shell (RSH) and remote execution (REXEC) are similar protocols that allow you to run programs and commands on different computers. The results are received and displayed on the local host. This can be useful for small computers to harness the power of large systems.

C.7.5 Domain Name System (DNS)

Recall that TCP/IP hosts are addressed by 32-bit IP addresses that are represented in decimal notation. For example, to Telnet to a remote hos with IP address of 9.67.38.1, the users would typically enter `telnet 9.67.38.1`. This approach can be cumbersome and error-prone, so a method was developed to use symbolic high-level machine names that are more meaningful to users than IP addresses.

This introduced the problem of maintaining the mappings between IP addresses and high-level machine names in a coordinated and centralized way. Initially, host names to address mappings were maintained by the Internet Network Information Center (InterNIC, previously NIC) in a single file (HOSTS.TXT) which was fetched by all hosts using FTP. Most hosts would have a copy of that file, which may or may not have been current or correct.

Due to the explosive growth in the number of hosts, this mechanism became too complicated and time-consuming, and was replaced by a new concept: the domain name system (DNS).

The domain concept lies in decentralizing the naming mechanism by distributing responsibility (and authority) for mapping between names and addresses. For example, consider the internal structure of a large organization. As the chief executive cannot do everything, the organization will probably be partitioned into divisions, each of them having autonomy within certain limits. Specifically, the executive in charge of a division has authority to make direct decisions, without permission from his chief executive.

Domain names are formed in a similar way, and will often reflect the hierarchical delegation of authority used to assign them. For example, consider the name

`small.itso.raleigh.ibm.com`

Here, itso.raleigh.ibm.com is the lowest-level domain name, a subdomain of raleigh.ibm.com, which again is a subdomain of ibm.com, a subdomain of com. We can also represent this naming concept by a hierarchical tree (see Figure 177 on page 225).

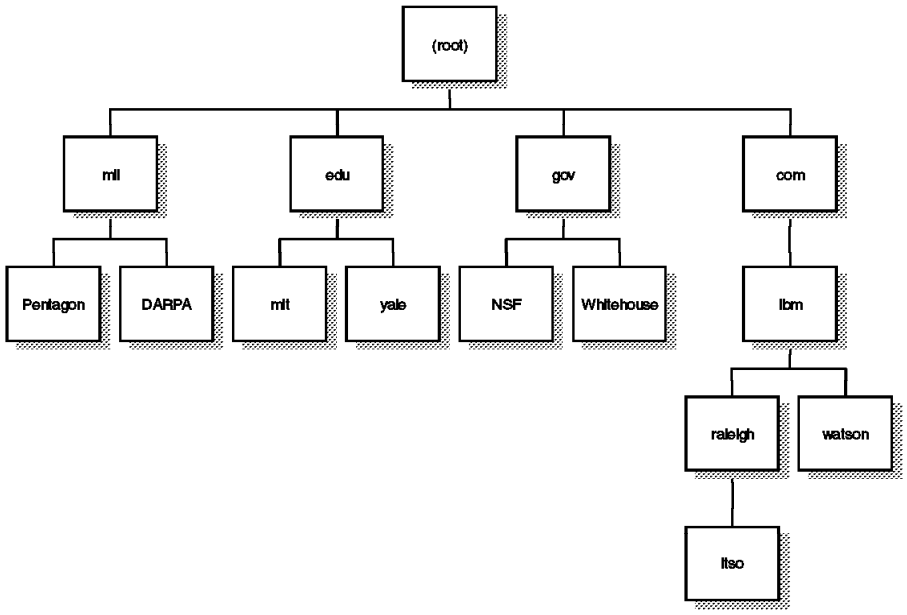


Figure 177. Hierarchical Namespace. (Chain of authority in assigning domain names)

Table 9 shows some of the top-level domains of today's Internet domain namespace.

Table 9. DNS - Some Top-Level Internet Domains

Domain Name	**Meaning**
com	Commercial organizations
edu	Educational institutions
gov	Government institutions
int	International organizations
mil	US Military
net	Major network support centers
org	Non-profit organizations
country code	ISO standard 2-letter identifier for country-specific domains

C.7.5.1 Mapping Domain Names to IP Addresses

The mapping of names to addresses consists of independent, cooperative systems called name servers. A name server is a server program that holds a master or a copy of a name-to-address mapping database, or otherwise points to a server that does, and that answers requests from the client software, called a name resolver.

Conceptually, all Internet domain servers are arranged in a tree structure that corresponds to the naming hierarchy in Figure 177. Each leaf represents a name server that handles names for a single subdomain. Links in the

conceptual tree do not indicate physical connections. Instead, they show which other name server a given server can contact.

Figure 178 shows the domain name resolution process that is summarized in the following steps:

1. A user program issues a request such as the gethostbyname() sockets call. (This particular call is used to ask for the IP address of a host by passing the hostname.)
2. The resolver formulates a query to the name server. (Full resolvers have a local name cache to consult first, stub resolvers do not.)
3. The name server checks to see if the answer is in its local authoritative database or cache, and if so, returns it to the client. Otherwise, it will query other available name server(s), starting down from the root of the DNS tree or as high up the tree as possible.
4. The user program will finally be given a corresponding IP address (or host name, depending on the query) or an error if the query could not be answered. Normally, the program will not be given a list of all the name servers that have been consulted to process the query.

The query/reply messages are transported by either UDP or TCP. DNS is conceptually defined in RFC 1034 and RFC 1035.

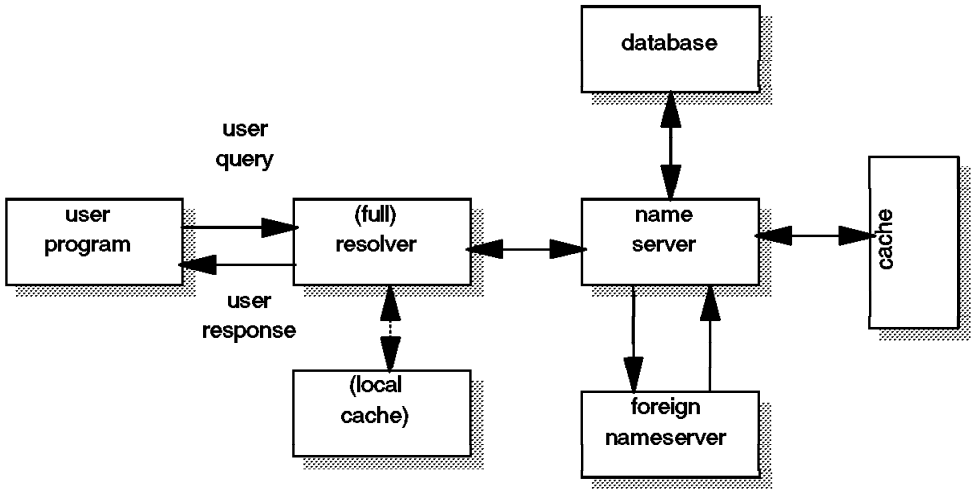


Figure 178. DNS - Resolver and Domain Name Server

C.7.5.2 Reverse Mapping

In some cases, it may be necessary to find a hostname for a given IP address. This is called reverse mapping and is a standard function of most DNS servers available today. A special domain, in-addr.arpa, is being used for inverse name queries.

C.7.5.3 Dynamic DNS (DDNS)

The Dynamic Domain Name System (DDNS) is a protocol that defines extensions to the Domain Name System to enable DNS servers to accept requests to add, update and delete entries in the DNS database dynamically. Because DDNS offers a functional superset to existing DNS servers, a DDNS server can serve both static and dynamic domains at the same time, a welcome feature for migration and overall DNS design.

DDNS is currently available in a non-secure and a secure flavor, defined in RFC 2136 and RFC 2137, respectively. Rather than allowing any host to update its DNS records, the secure version of DDNS uses public key security and digital signatures to authenticate update requests from DDNS hosts. IBM, for instance, has fully implemented secure DDNS on its OS/2 Warp Server, AIX, OS/390 and AS/400 platforms as well as on Windows NT.

Without client authentication, another host could impersonate an unsuspecting host by remapping the address entry for the unsuspecting host to that of its own. Once the remapping occurs, important data, such as logon passwords and mail intended for the host would unfortunately be sent to the impersonating host instead.

Please see also C.8.2.1, “Dynamic IP” on page 230 for more information on how DDNS works together with DHCP to perform seamless updates of reverse DNS mapping entries.

C.7.6 Simple Mail Transfer Protocol (SMTP)

The Simple Mail Transfer Protocol is an electronic mail protocol with both client (sender) and server (receive) functions. Since SMTP is a rather old protocol, many aspects of modern electronic mail are missing in its definitions. It basically assumes that messages would only consist of plain text in 7-bit US ASCII format with a line length of no more than 1000 characters. For more information about SMTP see RFCs 821, 822 and 974.

C.7.7 Multipurpose Internet Mail Extensions (MIME)

To overcome the shortcomings of SMTP, a new architecture has been defined that allows for a much greater variety of what can be contained in an electronic message, such as:

- 8-bit text and lines longer than 1000 characters
- International code pages and character sets
- Binary and multimedia objects, such as
  - Fonts
  - Images, audio and video objects

MIME is defined in RFCs 2045 to 2049 and currently has state of draft standard. MIME does not solely apply to electronic mail, it rather defines a way to incorporate different objects in any electronic message. For instance, it is used widely throughout the Internet today by means of browsing the World Wide Web (see C.10.3, “The World Wide Web (WWW)” on page 235).

C.7.8 Post Office Protocol (POP)

The post office protocol is an electronic mail protocol with both client (sender/receiver) and server (storage) functions. POP allows mail for multiple users to be stored in a central location until a request for delivery is made by an electronic mail program. A POP client typically has only one server mailbox, and all messages it retrieves are stored locally and then deleted at the server thus using server resources efficiently. For more information about POP, see RFC 1725.

C.7.9 Internet Message Access Protocol Version 4 (IMAP4)

IMAP4 is an electronic messaging protocol with both client and server functions. Similar to POP, IMAP4 servers store messages for multiple users to be retrieved upon client request, but IMAP4 clients have more capabilities in doing so than POP clients. IMAP4 allows clients to have multiple remote mailboxes to retrieve messages from and to choose any of those any time. IMAP4 clients can specify criteria for downloading messages, such as not to transfer large messages over slow links. Also, IMAP4 always keeps messages on the server and replicates copies to the clients. Transactions performed by disconnected clients are effected on server mailboxes by periodic re-synchronization of client and server. For more information on IMAP4 and its underlying electronic mail models, please see RFC 2060 and RFC 1733.

C.7.10 Remote Procedure Call (RPC)

Remote Procedure Call is a standard developed by SUN Microsystems and used by many vendors of UNIX systems.

RPC is an application programming interface (API) available for developing distributed applications. It allows programs to call subroutines that are executed at a remote system. The caller program (called client) sends a call message to the server process, and waits for a reply message. The call message includes the procedure′s parameters and the reply message contains the procedure′s results. RPC also provides a standard way of encoding data in a portable fashion between different systems called External Data Representation (XDR).

The concept of RPC is very similar to that of an application program issuing a procedure call:

- The caller process sends a call message and waits for the reply.
- On the server side, a process is dormant awaiting the arrival of call messages. When one arrives, the server process extracts the procedure parameters, computes the results and sends them back in a reply message.

C.7.10.1 Portmap

The Portmap or Portmapper is a server application that will map a program number and its version number to the Internet port number used by the program. Portmap is assigned the reserved (well-known service) port number 111.

Portmap only knows about RPC programs on the host it runs on. In order for Portmap to know about the RPC program, every RPC program should register itself with the local Portmapper when it starts up.

The RPC client (caller) has to ask the Portmap service on the remote host about the port used by the desired server program.

C.7.11 Network File System (NFS)

The Network File System (NFS) enables machines to share file systems across a network. It allows authorized users to access files located on remote systems as if they were local. It is designed to be machine-independent, operating system-independent, and transport protocol-independent. This is achieved through implementation on top of RPC.

C.7.12 X Window System

The X Window System (hereafter referred to as X) is one of the most widely used graphical user interface (GUI), or bitmapped-window display systems.

Current X releases contain two numbers: A version number indicating major protocol or standards revisions, and a release number indicating minor changes. At the time of this book's publication, the latest version is X11 Release 6, also known as X11R6.

There are two main components in X that communicate with each other:

C.7.12.1 X-Server

A dedicated program that provides display services on a graphic terminal, on behalf of a user, at the request of the user's X-client program. It controls the screen and handles the keyboard and the mouse (or other input devices) for one or more X-clients. Equally, it is responsible for output to the display, the mapping of colors, the loading of fonts and the keyboard mapping. Typically X-server programs run on high-performance graphics PCs and workstations, as well as X terminals, which are designed to run only the X-server program.

C.7.12.2 X-Client

The actual application, designed to employ a graphical user interface to display its output. Typically, many X-clients compete for the service of one X-server per display per user. Xterm and Xclock are two examples of X-clients.

C.7.12.3 Transport

The X Window System uses sockets to communicate over a TCP/IP network.

C.8 TCP/IP Configuration and Management Protocols

This section provides a brief overview of some of the protocols used to configure and manage TCP/IP networks.

C.8.1 Bootstrap Protocol (BOOTP)

The BOOTstrap Protocol (BOOTP) enables a client workstation to initialize with a minimal IP stack and request its IP address, a gateway address and the address of a name server from a BOOTP server. Once again, a good example of a client that requires this service is a diskless workstation. If BOOTP is to be used in your network, then you must make certain that both the server and client are on the same physical LAN segment. BOOTP can only be used across bridged segments when source-routing bridges are being used, or across subnets if you have a router capable of BOOTP forwarding, also known as a BOOTP relay agent.

C.8.2 Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) is based on BOOTP and extends the concept of a central server supplying configuration parameters to hosts in the network. DHCP adds the capability to automatically allocate reusable network addresses to workstations or hosts, and it supports the following functions:

Automatic Allocation
DHCP assigns a permanent address to a host.

Dynamic Allocation
DHCP assigns a leased IP address for a limited period of time. This is the only mechanism that allows automatic reuse of addresses that had been previously assigned but are no longer in use.

Manual Allocation
The host's address is manually configured by the network administrator.

You may have more than one DHCP server in your network, each server containing a pool of addresses and leases in local storage. However, you must not assign overlapping pools of addresses to several servers because, due to the lack of a server-to-server protocol, that would result in duplicate IP addresses being handed out by those servers. A client may be configured to broadcast a request for address assignment and will select the most appropriate response from those servers that answer the request. One big potential advantage with DHCP is a reduction in the workload required to manually configure addresses for all workstations in a segment.

A DHCP server does not need to be in the same subnet or on the same physical segment as the client which would then require the use of a BOOTP relay agent. More information about DHCP can be found in RFC 2031 and RFC 2032.

C.8.2.1 Dynamic IP

Using DHCP makes IP address assignment efficient and simple compared to previous practices. However, a change in a hosts IP address will in most cases cause problems for applications that rely on hostnames rather than IP addresses because DNS servers have no idea of DHCP and therefore any change to a host's IP address would have to be manually applied to DNS server databases by an administrator.

As mentioned in C.7.5.3, "Dynamic DNS (DDNS)" on page 227, ways now exist to dynamically update the DNS database as well. Dynamic IP, a term coined by IBM in 1995, closes the gap between DHCP and DDNS in the following way:

1. A client obtains IP address and configuration information from a DHCP server in the normal way.
2. If the client is DDNS-enabled and started for the first time, it will prompt the user for a hostname, create its security keys, register with its primary DDNS server, and then create its DNS records for name to IP mapping. It will then send a lease renewal request to the DHCP server and also supply its hostname so that the DHCP server can update the DNS records for reverse mapping.
3. If the client is DDNS-enabled and restarts, it will send its hostname with the initial DHCP request or discover message so that update of reverse records can be done right away. After obtaining a lease from the server, the DDNS client will then also update the name to IP mapping records.

4. If the client is not DDNS-enabled but supplies a hostname in its DHCP request and discover messages, the DHCP server can be configured to perform name to IP mapping updates to the DDNS server on behalf of that client in addition to the reverse mapping updates. That function is referred to as proxy-DDNS.

In any case, the lifetime of DDNS records for a particular host is tied to the lease time of the IP address assigned to that host by a DHCP server.

C.8.3 Simple Network Management Protocol (SNMP)

With the growth in size and complexity of the TCP/IP-based networks, the need for network management became very important. The current network management framework for TCP/IP consists of the following:

1. SMI (Structure and Identification of Management Information) describes how managed objects contained in the MIB (Management Information Base) are defined.
2. MIB-II (Management Information Base, second version) describes the managed objects.
3. SNMP defines the protocol used to manage these objects.

A network management station executes network management applications that monitor and control network elements such as hosts, gateways and terminal servers. These network elements use a management agent to perform the network management functions requested by the network management stations. SNMP is used to communicate management information between the network management stations and the agents in the network elements.

C.8.4 Lightweight Directory Access Protocol (LDAP)

Any information system relies on some sort of repository to retrieve locations of information, such as data, resources, addresses, and so forth. Those repositories are referred to as directories, much like a telephone directory, without which information may not easily be found, if at all, and communication in distributed systems would be impossible. Directories are typically used to hold user information (for authentication purposes), resource locations (for information access), and a whole lot more.

Though directories may contain all sorts of things and every vendor may implement its own style (in fact, many have done so), the Internet more or less demands some sort of directory standard to maintain coherent information and communication among diverse systems, applications, and vendors. The directory standard of choice today is X.500, as adopted by the International Standards Organization (ISO).

The more complex a directory becomes, the more overhead and cost may be involved in accessing it. LDAP specifies a simplified way to retrieve information from an X.500-compliant directory in an asynchronous, client/server type of protocol. For more information about LDAP and X.500 implementations, please see RFC 1777 and RFC 2116.

C.9 TCP/IP Routing Protocols and Techniques

This section provides a brief overview of some of the protocols used to update routing tables among routers in TCP/IP networks. This process is called dynamic routing because the routers take care that updates are sent automatically according to protocol specifications.

In contrast to that, static routing would require system administrators to enter all required routing information at every host and gateway in order for a TCP/IP internetwork to function as desired.

C.9.1 Routing Information Protocol (RIP)

The Routing Information Protocol creates and dynamically maintains network routing tables. RIP arranges to have gateways and routers periodically broadcast their routing tables to neighbors. Using this information, a RouteD server can update a host′s routing tables. For example, RouteD determines if a new route has been created, if a route is temporarily unavailable or if a more efficient route exists. For more information about routing using RIP, see RFC 1058.

The Routing Information Protocol Version 1 is commonly known as RIP. It uses a distance vector algorithm, which means it calculates the best path to a destination based on the number of hops in the path. Each hop represents a router through which a datagram must pass in order to reach the destination.

RIP is widely used and easy to implement, but it is known to have several limitations:

- The maximum number of hops is 15 (16 refers to an unreachable destination), making RIP inadequate for large networks that may have more than 15 routers on any single path.
- RIP is not a secure protocol. It does not authenticate the source of any routing updates it receives.
- RIP can not choose the best path based on delay, cost, reliability or load.
- RIP does not support variable length subnet masks.
- RIP can take a relatively long time (compared to other protocols such as OSPF) to converge, or stabilize its tables after an alteration to the network configuration has occurred.

The Routing Information Protocol Version 2 (RIP-2) was created in order to fix some of the limitations of RIP. It is still less powerful than protocols such as OSPF, but it has the advantages of being easy to implement and having a lower network overhead. This overhead includes network traffic and CPU time. RIP-2 can interoperate with RIP, and it also supports variable length subnetting and IP multicasting.

C.9.2 Open Shortest Path First (OSPF)

OSPF is a complex protocol utilizing a link state, shortest path first algorithm. In a link-state protocol, each router broadcasts link status information to each of its neighboring routers instead of distance vector information. Each neighboring router then propagates the status information to its own neighbors until the information has been sent to every router in the network. Each router then uses the status information to build a complete routing table utilizing a calculated cost for each link based on load, time delays, or reliability.

The biggest advantage of OSPF in comparison to either RIP or RIP-2 is that of the time taken to converge after a change to the network. The link-state protocols will always stabilize the propagated routing tables much faster than the distance vector protocols.

OSPF supports variable length subnetting and IP multicasting. It also introduces the concept of Areas, where the Autonomous System is divided into Areas, each responsible for its own topology. Area topology is not propagated to other Areas; border routers maintain connectivity between the separate Areas across an OSPF backbone, reducing the amount of routing information which must be exchanged.

See RFC 1583 for more information about OSPF.

C.9.3 Classless Inter-Domain Routing (CIDR)

It has been mentioned in C.5.1.1, “IP Addressing” on page 203 that, due to the impacts of growth, the IP address space is nearing exhaustion very soon if addresses were assigned as they are requested or as they used to be assigned previously. We have pointed out that the next version of IP, IPv6, will easily overcome that problem (see C.5.5, “The Future Version of IP (IPv6)” on page 213), but what can be done until IPv6 will be fully deployed?

One idea was to use a range of class C addresses instead of a single class B address. The problem there is that each network must be routed separately because standard IP routing understands only class A, B and C network addresses (see C.5.1.4, “IP Routing” on page 208).

Within each of these types of network, subnetting can be used to provide better granularity of the address space within each network, but there is no way to specify that multiple class C networks are actually related (see C.5.1.2, “IP Subnets” on page 205). The result of this is termed the *routing table explosion* problem: A class B network of 3000 hosts requires one routing table entry at each backbone router, whereas the same network, if addressed as a range of class C networks, would require 16 entries.

The solution to this problem is a scheme called Classless Inter-Domain Routing (CIDR). CIDR is described in RFCs 1518 to 1520.

CIDR does not route according to the class of the network number (hence the term classless) but solely according to the high order bits of the IP address which are termed the IP prefix. Each CIDR routing table entry contains a 32-bit IP address and a 32-bit network mask, which together give the length and value of the IP prefix. This can be represented as <IP_address network_mask>. For example, to address a block of 8 class C addresses with one single routing table entry, the following representation would suffice: <192.32.136.0 255.255.248.0>. This would, from a backbone point of view, refer to the class C network range from 192.32.136.0 to 192.32.143.0 as one single network because of the identical IP prefix, as illustrated in Figure 179 on page 234:

```
 11000000 00100000 10001000 00000000 = 192.32.136.0 (class C address)
 11111111 11111111 11111--- --------   255.255.248.0 (network mask)
===================================  logical_AND
 11000000 00100000 10001--- -------- = 192.32.136  (IP prefix)

 11000000 00100000 10001111 00000000 = 192.32.143.0 (class C address)
 11111111 11111111 11111--- --------   255.255.248.0 (network mask)
===================================  logical_AND
 11000000 00100000 10001--- -------- = 192.32.136  (same IP prefix)
```

Figure 179. Classless Inter-Domain Routing - IP Supernetting Example

This process of combining multiple networks into a single entry is referred to as supernetting because routing is based on network masks that are shorter than the natural network mask of an IP address, in contrast to subnetting (see C.5.1.2, “IP Subnets” on page 205) where the subnet masks are longer than the natural network mask.

CIDR is implemented and used in today’s Internet backbone routers based on the Border Gateway Protocol (BGB-4). It is scarcely used at the local network level where splitting up the available address space is more of a problem than expanding the address space.

Note: CIDR in itself does not constitute a routing protocol. It is a method of interpretation of IP addresses that can be employed by routing protocols to achieve the goals previously described.

C.10 Internet User Applications and Protocols

This section provides a brief overview of some of the protocols and applications that have made the task of using the Internet both easier and very popular over the past couple of years.

C.10.1 Network News

One application that is particularly popular on the Internet is Network News, also known as Usenet News. Based on the Network News Transfer Protocol (NNTP), users on the Internet can view and contribute to news groups covering topics such as science, education, computers, business, politics, recreation, sports, and many more. News groups are stored on news servers. NNTP is used for both server-to-server and client-to-server communication.

Clients use a news agent application, such as the IBM NewsReader/2, to retrieve articles from one or more news groups, and to post articles to one or more news groups. For more information about NNTP, see RFC 977.

C.10.2 Gopher

Gopher is a client/server protocol designed for information location and retrieval. The client function provides a menu-driven interface to access the files stored on a Gopher server. The server function allows descriptive names to be assigned to the files. Thus, making it easier to identify the content of each file. Gopher was designed at the University of Minnesota. For more information about Gopher, see RFC 1436.

C.10.3 The World Wide Web (WWW)

The World Wide Web is a global hypertext system that was initially developed in 1989 by Tim Berners Lee at the European Laboratory for Particle Physics, CERN in Switzerland to facilitate an easy way of sharing and editing research documents among a geographically dispersed group of scientists.

In 1993 the Web started to grow rapidly which was mainly due to the NCSA (National Center for Supercomputing Applications) developing a Web browser program called Mosaic, an X Windows-based application. This application provided the first graphical user interface to the Web and made browsing more convenient.

Today there are Web browsers and servers available for nearly all platforms. You can get them either from an FTP site for free or buy a licensed copy. The rapid growth in popularity of the Web is due to the flexible way people can navigate through world-wide resources in the Internet and retrieve them. To get an idea of the growth of the Web, Table 10 presents some statistics.

The number of Web servers is also increasing rapidly and the traffic over port 80, which is the well known port for HTTP Web servers, on the NSF backbone has had a phenomenal rate of growth too. The NSFNET was converted back to a private research network in 1995, therefore comprehensive statistics of backbone traffic are not as easily available anymore, if they are at all.

Table 10. Growth of the World Wide Web. The source of those figures can be found at the following URL:

`http://www.mit.edu/people/mkgray/net/web-growth-summary.html`

Date	Web Sites	Web Traffic	FTP Traffic	E-mail Traffic
June 1993	130	0.5	42.9	6.4
June 1994	2,738	6.1	35.2	6.4
March 1995	n/a	23.9	24.2	4.9
June 1995	23,500	n/a	n/a	n/a
June 1996	230,000	n/a	n/a	n/a
January 1997	650,000	n/a	n/a	n/a

C.10.4 Hypertext Transfer Protocol (HTTP)

The hypertext transfer protocol is a protocol designed to allow the transfer of hypertext markup language (HTML) documents. HTML is a tag language used to create hypertext documents. Hypertext documents include links to other documents that contain additional information about the highlighted term or subject. Such documents may contain other elements apart from text, such as graphic images, audio and video clips, and even virtual reality worlds (which are described in VRML, scripting language for that kind of elements).

HTTP is based on request-response activity. A client, running an application called a browser, establishes a connection with a server and sends a request to the server in the form of a request method. The server responds with a status line, including the message's protocol version and a success or error code, followed by a message containing server information, entity information and possible body content.

An HTTP transaction is divided into four steps:

1. The browser opens a connection.
2. The browser sends a request to the server.
3. The server sends a response to the browser.
4. The connection is closed.

On the Internet, HTTP communication generally takes place over TCP connections. The default port is TCP 80, but other ports can be used. This does not preclude HTTP from being implemented on top of any other protocol on the Internet, or on other networks. HTTP only presumes a reliable transport; any protocol that provides such guarantees can be used, but the mapping of the HTTP request and response structures onto the transport data units of the protocol in question is outside the scope of this document.

Except for experimental applications, current practice requires that the connection be established by the client prior to each request and closed by the server after sending the response. Both clients and servers should be aware that either party may close the connection prematurely, due to user action, automated timeout, or program failure, and should handle such closing in a predictable fashion. In any case, the closing of the connection by either or both parties always terminates the current request, regardless of its status.

What we have just described means that, in simple terms, HTTP is a connectionless protocol. To load a page including two graphics for example, a graphic-enabled browser will open three TCP connections: One for the page, and two for the graphics. Most browsers, however, are able to handle several of these connections simultaneously.

This behavior can be rather resource-intensive if one page consists of a lot of elements as quite a number of Web pages nowadays do. HTTP 1.1, as defined in RFC 2068, alleviates this problem to the extent that one TCP connection will be established per type of element on a page, and all elements of that kind will be transferred over the same connection.

HTTP is stateless, because it keeps no track of the connections. If a request depends on the information exchanged during a previous connection, then this information has to be kept outside the protocol.

C.10.5 The Advent of Java

Java is an important new technology in the world of the Internet. In summary, it is a simple, robust, object-oriented, platform-independent, multithreaded, dynamic general-purpose programming environment for creating applications for the Internet and intranet. Java includes the following components:

C.10.5.1 Java Language

Java is a programming language developed by Sun Microsystems, which is object-oriented, distributed, interpreted, architecture neutral, and portable. Java can be used to create downloadable program fragments, so-called applets, that augment the functionality of a Java-capable browser such as HotJava or Netscape Navigator.

C.10.5.2 Java Virtual Machine

The Java Virtual Machine (JVM) is an abstract computer that runs compiled Java programs (or precisely: that interprets Java byte-code that has been produced by a Java compiler). JVM is virtual because it is generally implemented in software on top of a real hardware platform and operating system. In this way, it is architecture-neutral and platform-independent. All Java programs should be compiled to run in a JVM.

The following diagram describes in simple terms how Java is implemented:

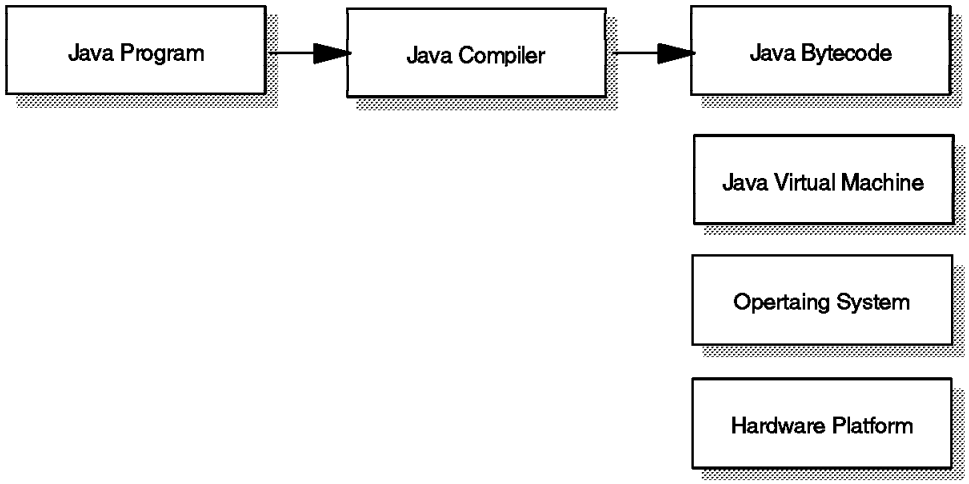


Figure 180. Implementation of Java

C.10.5.3 Programs, Applets and Servlets

When a Java program is started from inside an HTML (Web) page, it is called a Java applet as opposed to a Java program that is executed from the command line or otherwise on the local system. Applets are downloaded via the Web browser from a server and, by definition, are somewhat limited in the way they can use resources of the local system.

Originally, no Java applet was supposed to touch anything locally outside of its JVM and could only communicate back to the server where it was downloaded from. With Java 1.1, applets can be signed with security keys and certificates and can therefore be authenticated. Thus, an applet can be authorized to access local resources, such as file systems, and it may communicate with other systems.

In order to spare resources on clients and networks, Java applets can be executed on the server rather than downloaded and started at the client. Such programs are then referred to as servlets. Though that method requires a significantly more powerful server, it is highly suitable for environments with medialess systems, such as Network Computers.

C.10.5.4 HotJava

HotJava is a Java-enabled Web browser, developed by Sun Microsystems, which lets you view Java applets.

C.10.5.5 JavaOS

JavaOS is a highly compact operating system, developed by JavaSoft, which is designed to run Java applications directly on microprocessors in anything from personal computers to pagers. JavaOS will run equally well on a network computer, a PDA, a printer, a game machine, or countless other devices that require a very compact OS and the ability to run Java applications.

C.10.5.6 Java Beans

An initiative called Java Beans is brewing a similar set of APIs that will make it easy to create Java applications from reusable components. Java Beans will be used in a wide range of applications, from simple widgets to full-scale, mission-critical applications. Many software vendors including IBM have announced plans to support it.

C.10.5.7 JavaScript

JavaScript is an HTML extension and programming language, developed by Netscape, which is a simple object-based language compatible with Java. JavaScript programs are embedded as source directly in an HTML document. They can control the behavior of forms, buttons and text elements. It is used to create dynamic behavior in elements of the Web page. In addition, it can be used to create forms whose fields have built-in error checking routines.

For more information about Java, check out the following URLs:

`http://ncc.hursley.ibm.com/javainfo/`
`http://java.sun.com/`

C.11 TCP/IP and Internet Security

In this section we briefly introduce some concepts and protocols that allow you to establish various degrees of security in TCP/IP networks.

One may say that the Internet is great because there is so much information out there that can be accessed very easily and quickly. Electronic communication has become a lot easier because of the Internet, no doubt, but it can also be a dangerous thing at times.

Imagine that someone would get into your system and destroy data at random just because you forgot to implement security that could have prevented it. Or, worse, imagine someone would tap into your communication, learn your passwords and then use your account information to do electronic shopping.

One of the major concerns when providing commercial services on the Internet is providing for transaction security and communications security.

Information exchanges are secure if all the following are true:

- Messages are confidential.
- The information exchange has integrity.
- Both sender and receiver are accountable.
- You can authenticate both parties in the exchange.

There are certainly other ways of compromising information on the Internet that one might think of; so how should one go ahead to protect oneself against them?

C.11.1 Secure Sockets Layer (SSL)

SSL is a security protocol that was developed by Netscape Communications Corporation, along with RSA Data Security, Inc. The primary goal of the SSL protocol is to provide a private channel between communicating applications which ensures privacy of data, authentication of the partners and integrity.

SSL provides an alternative to the standard TCP/IP socket API which has security implemented within it. Hence, in theory it is possible to run any TCP/IP application in a secure way without changing it. In practice, SSL is so far only implemented for HTTP connections.

In fact the protocol is composed of two layers:

- At the lower layer is the SSL Record protocol. It is used for data encapsulation.
- On the upper layer is the SSL Handshake protocol used for initial authentication and transfer of encryption keys.

The SSL protocol addresses the following security issues:

Privacy
After the symmetric key is established in the initial handshake, the messages are encrypted using this key.

Integrity
Messages contain a message authentication code ensuring the message integrity.

Authentication
During the handshake, the client authenticates the server using an asymmetric or public key.

SSL requires each message to be encrypted and decrypted and therefore has a high performance and resource impact. In addition, since only the server is authenticated, SSL is not suitable for applications, such as electronic banking, which require that the server authenticate their clients.

C.11.2 Firewalls

One way to deal with network security is the installation of a specialized server, a so-called firewall. Firewalls tend to be seen as a protection between the Internet and a private network. But generally speaking a firewall should be considered as a means to divide the world into two or more networks: One or more secure networks and one or more non-secure networks.

Imagine a company where all the departments are connected to the internal network, including sales, accounts, development and human resources departments. The administrator would like to be able to restrict access from the development department machines to the human resources department machines and from the sales department to the development department.

In order to provide maximum security, a good firewall design is paramount:

- Anything not explicitly permitted should default to denied.
- Increasing complexity leads to bugs, which lead to opportunities.

- The server should be kept in a physically secure environment.
- Provide extensive logging.
- Turn off known problems and non-essential daemons (applications and services).

Most of the firewalls available today offer one or more of the following services, some of which we briefly describe in the paragraphs following the list below:

- Filtering gateways
- Proxy application layer gateways
- Circuit layer gateways (SOCKS servers)
- Domain name server hiding
- Mail handling
- Auditing and logging

Multiple technologies are needed to provide capabilities and protection. The IBM eNetwork Firewall, for instance, is based on IBM′s technology and has been used for more than ten years to protect internal IBM IP networks.

C.11.2.1 Screening Filters

The screening filter looks at each IP packet flowing through it, controlling access to machines and/or ports in the private network and possibly limiting access from the private network to the Internet. Screening filters operate at the IP layer and cannot control access at the application layer.

C.11.2.2 Proxy Servers

Proxy servers are used to control access to or from the private network relaying only acceptable communications from known users.

Users in the private network can access an application, such as FTP, in the proxy server using their usual utilities (clients). Users authenticate themselves to the proxy server and can then access the application on the desired machine in the public network. Proxy servers can also be used from the public network to access applications in the private network, but this exposes login names and passwords to attackers in the public network.

C.11.2.3 SOCKS Servers

SOCKS servers are like proxy servers without the requirement for double connections. With SOCKS, users can benefit from secure communications without needing to be aware that it is happening.

Users have to use new versions of applications called SOCKSified clients. The SOCKSified client code directs its requests to the SOCKS port on the firewall. Sessions are broken at the firewall, as they are with proxy servers. With SOCKS, however, the connection to the destination application is created automatically once the user is validated.

Both the client and the SOCKS server need to have SOCKS code. The SOCKS server acts as an application-level router between the client and the real application server. SOCKS V4 is for outbound TCP sessions only. It is simpler for the private network user, but does not have secure password delivery so it is not intended for sessions between public network users and private network applications. SOCKS V5 provides for several authentication methods and can therefore be used for inbound connections as well, though one should be cautious with that. SOCKS V5 also supports UDP-based applications and protocols.

The majority of Web browsers are SOCKSified and you can get SOCKSified TCP/IP stacks for most platforms. For additional information, refer to RFC 1928, 1929, 1961, and the following URL:

http://www.socks.nec.com

C.11.3 IP Security Architecture (IPSec)

Several security mechanisms are being described that address security, authentication and encryption at the IP layer rather than on upper transport or application layers. The IP Security Architecture defines two specific headers to provide security services for IP datagrams. Those may be applied either separately or combined:

C.11.3.1 IP Authentication Header (AH)

The IP Authentication Header (AH) can be used to provide connectionless integrity and data origin authentication for IP datagrams, and optionally to provide anti-replay integrity. This header protects an entire IP datagram, including all immutable fields in the IP header. AH does not provide confidentiality (no encryption).

C.11.3.2 IP Encapsulated Security Payload (ESP)

The Encapsulating Security Payload (ESP) can be used to provide confidentiality (encryption), data origin authentication, connectionless integrity, anti-replay integrity, and limited traffic flow confidentiality. Unlike AH, ESP provides security only for the protocols encapsulated by it, not the protocol that carries it.

C.11.3.3 Key Management

IPSec uses shared secret keys to protect every single conversation (called security association). This means that a pair of keys must be exchanged between two systems before they can use IPSec for secure one-way communication, and two key pairs must be exchanged for secure two-way communication.

Because keys are different for each security association, this process is repetitive for any number of hosts that a single system wants to securely communicate with. Even in a small network, this is virtually unachievable, so the IETF has endorsed a standard key management framework called Internet Security Association Key Management Protocol (ISAKMP) and a key determination protocol called Oakley. Currently, IPSec key management is not yet defined as an Internet standard.

More information on IPSec can be found in RFCs 1825 to 1829. For Information on ISAKMP/Oakley, please refer to the following URLs:

http://ds.internic.net/internet-drafts/draft-ietf-ipsec-isakmp-08.txt
http://ds.internic.net/internet-drafts/draft-ietf-ipsec-oakley-02.txt
http://ds.internic.net/internet-drafts/draft-ietf-ipsec-isakmp-oakley-05.txt

C.11.4 Virtual Private Networks

We have discussed in C.3, “Internet Standards and Request for Comments (RFC)” on page 200 that whoever wants to communicate directly over the Internet must register with the InterNIC and obtain valid IP addresses. However, such addresses are high in demand and short in supply. Moreover, organizations as well as private citizens have an increasing interest in

communicating securely across the Internet which is known as a rather insecure area to work in.

It would therefore be desirable to extend both the confidentiality and address flexibility of intranets across the Internet. That can be achieved by creating secure IP tunnels between the endpoint of a private network and another such endpoint or a remote system, as illustrated in Figure 181.

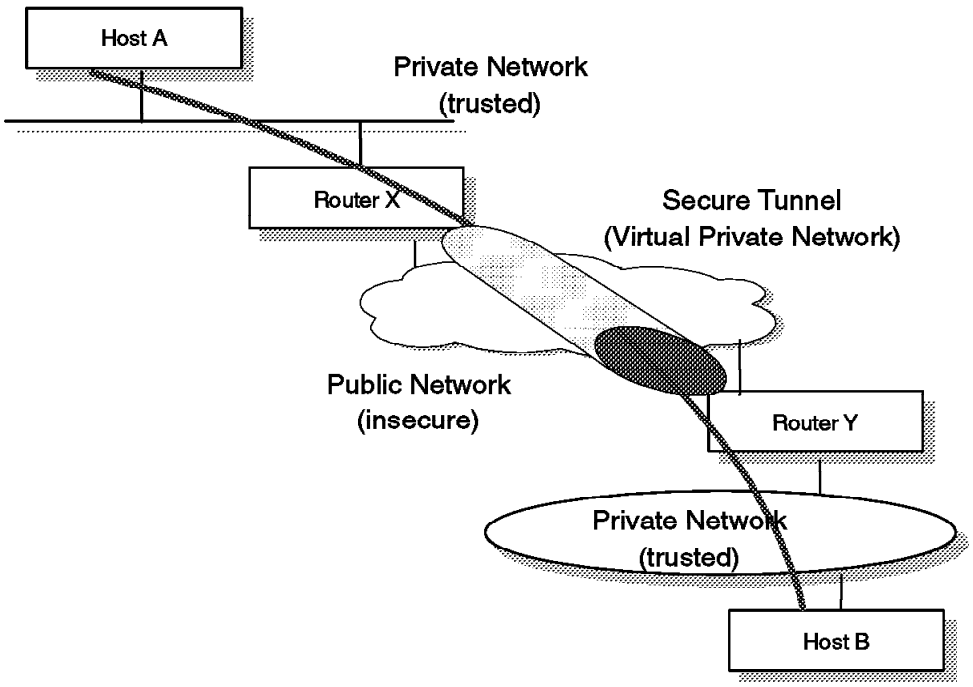


Figure 181. Secure IP Tunnels

IP tunneling can be described and used in the following way:

1. As traffic enters a tunnel, the original IP datagram is placed inside another IP datagram as a payload and optionally also encrypted.
2. Only the addresses at the endpoints of the tunnels are visible to the Internet and therefore have to be registered.
3. Addresses on remote networks are hidden from the outside by the tunnel and can be locally administered. This can be an effective way to prevent address spoofing attacks and effectively turns the Internet into a private network, hence the term Virtual Private Networks.
4. Traffic inside the tunnel can be encrypted and therefore totally hidden from snooping while traversing the Internet.

IPSec ESP provides a way of tunneling IP in a secure manner. Other tunneling protocols in use today are Point-to-Point Tunneling Protocol (PPTP), developed by Microsoft, 3Com and others, and Layer 2 Forwarding (L2F), developed by Cisco Systems. Based on these two protocols, the IETF is currently working on a standardized Layer 2 Tunneling Protocol (L2TP). As opposed to IPSec, PPTP, L2F and L2TP operate on the network interface layer and are therefore suited to

encapsulate and tunnel a variety of protocols. More information can be found at the following URLs:

```
http://ds.internic.net/internet-drafts/draft-ietf-ipsec-vpn-00.txt
http://ds.internic.net/internet-drafts/draft-ietf-pppext-pptp-02.txt
http://ds.internic.net/internet-drafts/draft-valencia-l2f-00.txt
http://ds.internic.net/internet-drafts/draft-ietf-pppext-l2tp-09.txt
```

C.12 Real-Time and Multimedia Application Support

Usually, the Internet does not provide a quality of service guarantee to the users. Routers normally forward packets on a first come first serve basis (unless specifically configured) and start discarding packets when their queues fill up. While such behavior could be tolerated in the past for protocols such as FTP, Telnet and HTTP, emerging applications such as real-time audio and video impose a much higher demand on network resources.

C.12.1 Resource Reservation Protocol (RSVP)

RSVP has been designed to support multicast applications such as video transmissions. In this case, a server would periodically announce that is has a pending transmission and provide some characteristics of that transmission. Potentially interested clients should then be able to determine the priority and bandwidth required throughout the network in order to be able to receive the data flow at appropriate speed.

All routers that have forwarded the server′s announcement and that support RSVP, should have made a note of it to allow the client to back-track the path to the source across the network. The client application then uses RSVP to send a resource reservation request back along that path to notify the server and all intermediate routers that it is interested in this transmission and that the required resources should be provided.

This still leaves a few problems to be solved:

1. If servers don′t announce characteristics of their pending or ongoing transmissions, the clients would not know what type of reservation to place.
2. All routers along the path, as well as server and client, must support RSVP.
3. It is yet undefined what should happen when the routers cannot handle any more priority traffic or bandwidth.

More information on RSVP can be found in RFC 2205.

C.12.2 Real-Time Protocol (RTP)

In order to facilitate reliable time-critical one-to-many or many-to-many transmissions, current transport protocols are no longer sufficient. While TCP only provides reliable one-to-one communication, UDP does not provide any time-critical flow information such as time stamps or sequence numbers.

RTP has been designed to assist in this case. It is not a transport protocol since it operates on top of UDP, but it adds both sequence numbers and time stamps to assure packet order and synchronization between senders and receivers.

Multicast applications can use RSVP to reserve bandwidth beforehand, or they can use special hosts to assist with transmission over low-bandwidth links, such as the following:

Translators

A translator converts a high quality (and therefore high bandwidth) payload format into a lower quality format that can be transmitted over a slower link. The number of streams to be sent remains the same.

Mixers

A mixer combines a number of streams into a single one that can be accommodated by a slower link.

Alongside RTP which provides only transmission of data goes the Real-Time Control Protocol (RTCP). RTCP is used to send feedback from everyone in a particular multicast group to that group to let everyone know how well a transmission is received and what the overall load on the network would be. Clients can make resource reservation requests based on that information. Servers, or senders, periodically transmit sender reports to inform receivers about what they should have received. More information on RTP can be found in RFC 1889.

C.13 Transporting Other Protocols over TCP/IP

We have so far regarded the TCP/IP network and transport protocols as a means of communications for TCP/IP applications. In reality, those protocols can also be employed to connect systems that run applications that normally would not have anything to do with TCP/IP. Those applications would expect to communicate over protocols such as NetBIOS, SNA or IPX.

Whenever an application or a protocol is made to use a transport protocol other than the one(s) it has originally been designed for, we call that non-native transport.

C.13.1 NetBIOS over TCP/IP

RFCs 1001 and 1002 define a way how to support applications using the NetBIOS API to use a TCP/IP network for transport. Essentially, NetBIOS is not a protocol but an application programming interface that knows very little about any underlying networking protocols. NetBIOS is used mostly in LAN environments and is there implemented to use the IEEE 802.2 interface. That renders NetBIOS unusable for TCP/IP networks because it cannot be routed in that way.

RFC 1001/1002 implementations of NetBIOS over TCP/IP alleviate that problem and have become increasingly important with the presence of operating systems such as OS/2 Warp Server, Windows NT and Windows 95 which partly rely on NetBIOS for communications.

C.13.2 SNA over TCP/IP

The IBM Multiprotocol Transport Network Architecture (MPTN) defines another way of using any transport network for any kind of application. The ultimate goal and great benefit of MPTN is that applications remain unchanged even if their native transport network is replaced by a non-native transport network.

In a specific implementation of MPTN, a TCP/IP network can be used to transport SNA applications using either dependent or independent logical unit (LU) communication, and it also support Advanced-Program-to-Program Communication (APPC) and Advanced-Peer-to-Peer Networking (APPN).

Another implementation, MPTN likewise supports the transport of TCP/IP sockets applications over SNA networks.

C.13.3 IPX over TCP/IP

In a similar way as with NetBIOS over TCP/IP (RFC 1001/1002), RFC 1234 describes a way to transport the Novell Internet Packet Exchange (IPX) protocol over IP. However, in this case IPX datagrams are encapsulated in UDP datagrams before being sent over an IP network. This makes a whole TCP/IP network appear as a single IPX network to NetWare servers and requesters.

IPX offers functions to NetWare servers and IPX routers that are similar to the functions which IP provides for TCP/IP networks. Therefore, a connectionless delivery over UDP is desired when sending IPX over IP.

C.14 TCP/IP and Internet Publications

For more detailed or advanced knowledge of TCP/IP, please refer to the following selected publications:

- *Internetworking with TCP/IP, Volume I, Principles, Protocols and Architecture*, third edition, Prentice-Hall, Inc., 1995, by Douglas E. Comer; ISBN 0-13-216987-8.
- *TCP/IP Tutorial and Technical Overview*, fifth edition, IBM Corp.,1995, GG24-3376-04, and Prentice-Hall, Inc., 1995, by Eamon Murphy, Steve Hayes, Matthias Enders; ISBN 0-13-460858-5.
- *IPng and the TCP/IP Protocols*, John Wiley & Sons, Inc., 1996, by Stephen A. Thomas; ISBN 0-471-13088-5.
- *Communications for Cooperating Systems - OSI, SNA and TCP/IP*, Addison-Wesley, Publishing Company, Inc., 1992, by R. J. Cypser; ISBN 0-201-50775-7.
- *The Request For Comments (RFCs)*

  There are more than 2200 RFCs today. For those readers who want to keep up-to-date with the latest advances and research activities in TCP/IP, the ever-increasing number of RFCs and Internet Drafts (ID) is the best source of this information. (See C.3, “Internet Standards and Request for Comments (RFC)” on page 200 for more details on RFCs.)

  RFCs can be viewed or obtained online from the IETF Web page using the following URL:

  `http://ds.internic.net/ds/dspg0intdoc.html`

Appendix D. Special Notices

This publication is intended to help systems programmers understand the new functions available with OS/390 Release 5 and assist with the install of this product. The information in this publication is not intended as the specification of any programming interfaces that are provided by OS/390 Release 5. See the PUBLICATIONS section of the IBM Programming Announcement for OS/390 Version 2 Release 5 for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these

names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ACF/VTAM®
AFP
AIX/ESA®
CBPDO
IBM®
IMS/ESA®
IP PrintWay
OS/2®
Parallel Sysplex
PrintWay
RACF
System/390®
VM/ESA®
Advanced Function Printing
AIX®
AIX/6000®
CICS®
IMS
Intelligent Printer Data Stream
NetSpool
OS/390
Print Services Facility
PSF
RMF
VisualAge®

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

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

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.

Appendix E. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

E.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 257.

Short Title	Title	Order Number
Stay Cool on OS/390: Installing Firewall Technology	*Stay Cool on OS/390: Installing Firewall Technology*	SG24-2046
Selecting a Server - The Value of OS/390	*Selecting a Server - The Value of OS/390*	SG24-4812
A Comparison of System/390 Configurations - Parallel and Traditional	*A Comparison of System/390 Configurations - Parallel and Traditional*	SG24-4514
OS/390 Release 4 Implementation	*OS/390 Release 4 Implementation*	SG24-2089
OS/390 Release 3 Implementation	*OS/390 Release 3 Implementation*	SG24-2067
OS/390 Release 2 Implementation	*OS/390 Release 2 Implementation MVS, SMP/E, SDSF, and RMF*	SG24-4834
Version 5 Implementation Guide	*MVS/ESA Version 5 Implementation Guide*	SG24-4584
HCD and Dynamic I/O Reconfiguration Primer	*MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*	SG24-4037
Sysplex Migration Guide	*MVS/ESA Version 5 Sysplex Migration Guide*	SG24-4581
S/390 G3 Enterprise Server: CSAR Presentation Guide	*S/390 G3 Enterprise Server: Complex Systems Availability and Recovery Presentation Guide*	SG24-4911

E.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
Lotus Redbooks Collection	SBOF-6899	SK2T-8039
Tivoli Redbooks Collection	SBOF-6898	SK2T-8044
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RS/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RS/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
RS/6000 Redbooks Collection (PDF Format)	SBOF-8700	SK2T-8043
Application Development Redbooks Collection	SBOF-7290	SK2T-8037

E.3 Other Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook. A publication whose order number begins with the prefix **LY** is available to IBM-licensed customers only.

Note: A few publications in the following list are no longer available in hardcopy format and cannot be separately ordered. These publications are only available on CD-ROMs.

These publications are also relevant as further information sources:

- OS/390 OpenEdition

Short Title	Title	Order Number
OS/390 OpenEdition MVS User's Guide	*OS/390 OpenEdition MVS User's Guide*	SC28-1891
OS/390 OpenEdition MVS Command Reference	*OS/390 OpenEdition MVS Command Reference*	SC28-1892
OS/390 OpenEdition MVS Programming Tools	*OS/390 OpenEdition MVS Programming Tools*	SC28-1904
OS/390 OpenEdition MVS Messages and Codes	*OS/390 OpenEdition MVS Messages and Codes*	SC28-1908
OS/390 OpenEdition MVS Programming: Assembler Callable Services Reference	*OS/390 OpenEdition MVS Programming: Assembler Callable Services Reference*	SC28-1899
OS/390 OpenEdition MVS Planning	*OS/390 OpenEdition MVS Planning*	SC28-1890
OS/390 OpenEdition MVS File System Interface Reference	*OS/390 OpenEdition MVS File System Interface Reference*	SC28-1909
OS/390 OpenEdition MVS Using REXX and OpenEdition MVS	*OS/390 OpenEdition MVS Using REXX and OpenEdition MVS*	SC28-1905
OS/390 OpenEdition MVS Communications Server Guide	*OS/390 OpenEdition MVS Communications Server Guide*	SC28-1906

- I/O Configuration Management

Short Title	Title	Order Number
OS/390 HCD Planning	*OS/390 Hardware Configuration Definition Planning*	GC28-1750
OS/390 HCD User's Guide	*OS/390 HCD User's Guide*	SC28-1848
HCD Messages	*OS/390 HCD Messages*	GC28-1849

- RMF

Short Title	Title	Order Number
RMF Messages and Codes	*OS/390 RMF Messages and Codes*	GC28-1948
RMF Performance Management Guide	*OS/390 RMF Performance Management Guide*	SC28-1951
RMF User's Guide	*OS/390 RMF User's Guide*	SC28-1949
RMF Report Analysis	*OS/390 RMF Report Analysis*	SC28-1950

Short Title	Title	Order Number
RMF Programmer's Guide	*OS/390 RMF Programmer's Guide*	SC28-1952

- Multi-System Configuration Management

Short Title	Title	Order Number
OS/390 Parallel Sysplex Systems Management	*OS/390 Parallel Sysplex Systems Management*	GC28-1861
OS/390 Parallel Sysplex Hardware and Software Migration	*OS/390 Parallel Sysplex Hardware and Software Migration*	GC28-1862
OS/390 Parallel Sysplex Application Migration	*OS/390 Parallel Sysplex Application Migration*	GC28-1863
OS/390 V2R5.0 MVS Setting Up a Sysplex	*OS/390 MVS Setting Up a Sysplex*	GC28-1779
OS/390 V2R5.0 MVS Sysplex Services Guide	*OS/390 MVS Programming: Sysplex Services Guide*	GC28-1771
OS/390 V2R5.0 MVS Sysplex Services Reference	*OS/390 MVS Programming: Sysplex Services Reference*	GC28-1772

- OS/390 Operating System

Short Title	Title	Order Number
OS/390 V2R5.0 MVS Auth Assembler Services Reference ALE-DYN	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 1, ALE-DYN*	GC28-1764
OS/390 V2R5.0 MVS Auth Assembler Services Reference ENF-ITT	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 2, ENF-ITT*	GC28-1765
OS/390 V2R5.0 MVS Auth Assembler Services Reference LLA-SDU	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 3, LLA-SDU*	GC28-1766
OS/390 V2R5.0 MVS Auth Assembler Services Reference SET-WTO	*OS/390 MVS Programming: Authorized Assembler Services Reference, Volume 4, SET-WTO*	GC28-1767
OS/390 V2R5.0 MVS Extended Addressability Guide	*OS/390 MVS Programming: Extended Addressability Guide*	GC28-1769
OS/390 V2R5.0 MVS Assembler Services Guide	*OS/390 MVS Programming: Assembler Services Guide*	GC28-1762
OS/390 V2R5.0 MVS Assembler Services Reference	*OS/390 MVS Programming: Assembler Services Reference*	GC28-1910
OS/390 V2R5.0 MVS Auth Assembler Services Guide	*OS/390 MVS Programming: Authorized Assembler Services Guide*	GC28-1763
Introducing OS/390	*OS/390 Introduction and Release Guide*	GC28-1725
OS/390 V1R2.0 MVS JCL User's Guide	*OS/390 MVS JCL User's Guide*	GC28-1758
OS/390 V2R5.0 MVS JCL Reference	*OS/390 MVS JCL Reference*	GC28-1757
OS/390 V2R5.0 MVS Callable Services for HLL	*OS/390 MVS Programming: Callable Services for High-Level Languages*	GC28-1768

Short Title	Title	Order Number
OS/390 V2R5.0 MVS Writing TPs for APPC/MVS	*OS/390 MVS: Writing Transaction Programs for APPC/MVS*	GC28-1775
OS/390 V2R5.0 MVS Planning: APPC/MVS Management	*OS/390 MVS Planning: APPC/MVS Management*	GC28-1807
OS/390 V2R5.0 MVS IPCS Commands	*OS/390 MVS Interactive Problem Control System (IPCS) Commands*	GC28-1754
OS/390 V2R5.0 MVS IPCS User's Guide	*OS/390 MVS Interactive Problem Control System (IPCS) User's Guide*	GC28-1756
OS/390 V2R5.0 MVS IPCS Customization	*OS/390 MVS Interactive Problem Control System (IPCS) Customization*	GC28-1755
OS/390 V2R5.0 MVS Initialization and Tuning Guide	*OS/390 MVS Initialization and Tuning Guide*	SC28-1751
OS/390 V2R5.0 MVS Initialization and Tuning Reference	*OS/390 MVS Initialization and Tuning Reference*	SC28-1752
OS/390 V2R5.0 MVS Installation Exits	*OS/390 MVS Installation Exits*	SC28-1753
OS/390 V2R5.0 MVS Conversion Notebook	*OS/390 MVS Conversion Notebook*	GC28-1747
OS/390 V2R5.0 MVS System Commands Summary	*OS/390 MVS System Commands Summary*	GX22-0040
OS/390 V2R5.0 Planning for Installation	*OS/390 Planning for Installation Release 4*	GC28-1726
OS/390 V2R5.0 MVS System Commands	*OS/390 MVS System Commands*	GC28-1781
OS/390 V2R5.0 MVS System Management Facilities (SMF)	*OS/390 MVS System Management Facilities (SMF)*	GC28-1783
OS/390 V2R5.0 MVS Planning: Operations	*OS/390 MVS Planning: Operations*	GC28-1760
OS/390 V2R5.0 MVS Planning: Global Resource Serialization	*OS/390 MVS Planning: Global Resource Serialization*	GC28-1759
OS/390 V2R5.0 MVS System Data Set Definition	*OS/390 MVS System Data Set Definition*	GC28-1782
OS/390 V2R5.0 MVS System Messages, Vol 1 (ABA-ASA)	*OS/390 MVS System Messages, Volume 1 (ABA-ASA)*	GC28-1784
OS/390 V2R5.0 MVS System Messages, Vol 2 (ASB-ERB)	*OS/390 MVS System Messages, Volume 2 (ASB-ERB)*	GC28-1785
OS/390 V2R5.0 MVS System Messages, Vol 3 (GDE-IEB)	*OS/390 MVS System Messages, Volume 3 (GDE-IEB)*	GC28-1786
OS/390 V2R5.0 MVS System Messages, Vol 4 (IEC-IFD)	*OS/390 MVS System Messages, Volume 4 (IEC-IFD)*	GC28-1787
OS/390 V2R5.0 MVS System Messages, Vol 5 (IGD-IZP)	*OS/390 MVS System Messages, Volume 5 (IGD-IZP)*	GC28-1788
OS/390 V2R5.0 MVS Dump Output Messages	*OS/390 MVS Dump Output Messages*	GC28-1749
OS/390 V2R5.0 MVS System Codes	*OS/390 MVS System Codes*	GC28-1780
OS/390 V2R5.0 MVS Routing and Descriptor Codes	*OS/390 MVS Routing and Descriptor Codes*	GC28-1778
OS/390 V2R5.0 MVS Recovery and Reconfiguration Guide	*OS/390 MVS Recovery and Reconfiguration Guide*	GC28-1777
OS/390 V2R5.0 MVS JES Common Coupling Services	*OS/390 MVS Programming: JES Common Coupling Services*	GC28-1770

Short Title	Title	Order Number
OS/390 V2R5.0 MVS: Writing Servers for APPC/MVS	*OS/390 MVS: Writing Servers for APPC/MVS*	GC28-1774
OS/390 V2R5.0: MVS Writing Transaction Schedulers for APPC/MVS	*OS/390 MVS: Writing Transaction Schedulers for APPC/MVS*	GC28-1776
OS/390 MVS APPC/MVS Handbook for OS/2	*OS/390 MVS APPC/MVS Handbook for OS/2*	GC28-1746
OS/390 V2R5.0 MVS Product Registration	*OS/390 MVS Programming: Product Registration*	GC28-1729
OS/390 V2R5.0 MVS Product Management	*OS/390 MVS Product Management*	GC28-1730
OS/390 V2R5.0 MVS Planning: Workload Management	*OS/390 MVS Planning: Workload Management*	GC28-1761
OS/390 V2R5.0 MVS Workload Management Services	*OS/390 MVS Programming: Workload Management Services*	GC28-1773
OS/390 Information Roadmap	*OS/390 Information Roadmap*	GC28-1727
OS/390 V2R5.0 MVS Diagnosis: Tools and Service Aids	*OS/390 MVS Diagnosis: Tools and Service Aids*	LY28-1845
OS/390 Introduction and Release Guide	*OS/390 Introduction and Release Guide*	GC28-1725
OS/390 Planning for Installation	*OS/390 V2R5.0 Planning for Installation*	GC28-1726
OS/390 Information Roadmap	*OS/390 V2R5.0 Information Roadmap*	GC28-1727

- SMP/E and Installation Manuals

Short Title	Title	Order Number
ServerPac Guide and Worksheet	*ServerPac Guide and Worksheet*	SC28-1244
MVS Packaging Rules	*Standard Packaging Rules for MVS-Based Products*	SC23-3695
OS/390 SMP/E Messages and Codes	*OS/390 System Modification Program Extended Messages and Codes*	SC28-1738
OS/390 SMP/E Command Reference	*OS/390 System Modification Program Extended Command Reference*	SC28-1805
OS/390 SMP/E Reference	*OS/390 System Modification Program Extended Reference*	SC28-1806

- JES2 Subsystem

Short Title	Title	Order Number
OS/390 JES2 Messages	*OS/390 JES2 Messages*	GC28-1796
OS/390 JES2 Commands	*OS/390 JES2 Commands*	GC28-1790
OS/390 JES2 Initialization and Tuning Guide	*OS/390 JES2 Initialization and Tuning Guide*	SC28-1791
OS/390 JES2 Initialization and Tuning Reference	*OS/390 JES2 Initialization and Tuning Reference*	SC28-1792
OS/390 JES2 Installation Exits	*OS/390 JES2 Installation Exits*	SC28-1793
OS/390 JES2 Macros	*OS/390 JES2 Macros*	SC28-1795

Short Title	Title	Order Number
OS/390 JES2 Migration Notebook	*OS/390 JES2 Migration Notebook*	GC28-1797

- JES3 Subsystem

Short Title	Title	Order Number
OS/390 JES3 Conversion Notebook	*OS/390 JES3 Conversion Notebook*	GC28-1799
OS/390 JES3 Initialization and Tuning Guide	*OS/390 JES3 Initialization and Tuning Guide*	SC28-1802
OS/390 JES3 Initialization and Tuning Reference	*OS/390 JES3 Initialization and Tuning Reference*	SC28-1803
OS/390 JES3 Messages	*OS/390 JES3 Messages*	GC28-1804
OS/390 JES3 Commands	*OS/390 JES3 Commands*	GC28-1798

- ICKDSF

Short Title	Title	Order Number
ICKDSF R16 Refresh User's Guide	*ICKDSF R16 Refresh User's Guide*	GC35-0033

- Security Server

Short Title	Title	Order Number
OS/390 Security Server (RACF) System Programmer's Guide	*OS/390 Security Server (RACF) System Programmer's Guide*	SC28-1913
OS/390 Security Server (RACF) Macros and Interfaces	*OS/390 Security Server (RACF) Macros and Interfaces*	SC28-1914
OS/390 Security Server (RACF) Command Language Reference	*OS/390 Security Server (RACF) Command Language Reference*	SC28-1919
OS/390 Security Server (RACF) Introduction	*OS/390 Security Server (RACF) Introduction*	GC28-1912
OS/390 Security Server (RACF) Messages and Codes	*OS/390 Security Server (RACF) Messages and Codes*	SC28-1918
OS/390 Security Server (RACF) Security Administrator's Guide	*OS/390 Security Server (RACF) Security Administrator's Guide*	SC28-1915
OS/390 Security Server (RACF) Auditor's Guide	*OS/390 Security Server (RACF) Auditor's Guide*	SC28-1916
OS/390 Security Server External Security Interface (RACROUTE) Macro Reference	*OS/390 Security Server External Security Interface (RACROUTE) Macro Reference*	GC28-1922
OS/390 Security Server (RACF) Planning: Installation and Migration	*OS/390 Security Server (RACF) Planning: Installation and Migration*	GC28-1920
OS/390 Security Server (RACF) Support for MVS OpenEdition DCE, SOMobjects for MVS and SystemView	*OS/390 Security Server (RACF) Support for MVS OpenEdition DCE, SOMobjects for MVS and SystemView*	GC28-1924
OS/390 Security Server (OpenEdition DCE Security Server) Overview	*OS/390 Security Server (OpenEdition DCE Security Server) Overview*	GC28-1938

- TSO/E

Short Title	Title	Order Number
TSO/E Administration	*OS/390 TSO/E Administration*	SC28-1966
TSO/E REXX Reference	*OS/390 TSO/E REXX Reference*	SC28-1975
TSO/E General Information	*OS/390 TSO/E General Information*	GC28-1964
TSO/E Customization	*OS/390 TSO/E Customization*	SC28-1965
TSO/E Programming Guide	*OS/390 TSO/E Programming Guide*	SC28-1970
TSO/E Programming Services	*OS/390 TSO/E Programming Services*	SC28-1971
TSO/E CLISTs	*OS/390 TSO/E CLISTs*	SC28-1973
TSO/E User's Guide	*OS/390 TSO/E User's Guide*	SC28-1968
TSO/E REXX User's Guide	*OS/390 TSO/E REXX User's Guide*	SC28-1974
TSO/E System Programming Command Reference	*OS/390 TSO/E System Programming Command Reference*	SC28-1972
TSO/E Command Reference	*OS/390 TSO/E Command Reference*	SC28-1969
TSO/E Messages	*OS/390 TSO/E Messages*	GC28-1978

- PSF/MVS, IP PrintWay, and NetSpool

Short Title	Title	Order Number
AFP Printer Information	*Advanced Function Presentation: Printer Information*	G544-3290
AFP Printer Summary	*Advanced Function Presentation: Printer Summary*	G544-3135
Guide to Advanced Function Presentation	*Guide to Advanced Function Presentation*	G544-3876
OS/390 Print Interface Configuration Guide	*OS/390 Print Interface Configuration Guide*	G544-5544
OS/390 Print Server User's Guide for OS/390 UNIX System Services	*OS/390 Print Server User's Guide for OS/390 UNIX System Services*	S544-5543
OS/390 Print Server User's Guide for Windows	*OS/390 Print Server User's Guide for Windows*	S544-5511
IBM IP PrintWay Guide	*IBM IP PrintWay Guide*	S544-5379
IBM NetSpool Guide	*IBM NetSpool Guide*	G544-5301
OS/390 V1R3.0: Printing Softcopy BOOKs	*OS/390 V1R3.0: Printing Softcopy BOOKs*	S544-5354
PSF/MVS Messages and Codes	*Print Services Facility/MVS: Messages and Codes Version 2 Release 2 Modification 0*	S544-3675

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 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:

- **Redbooks Web Site on the World Wide Web**

  http://w3.itso.ibm.com/

- **PUBORDER** — to order hardcopies in the United States
- **Tools Disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get BookManager BOOKs of redbooks, type the following command:

  ```
  TOOLCAT REDBOOKS
  ```

  To get lists of redbooks, type the following command:

  ```
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  ```

  To register for information on workshops, residencies, and redbooks, type the following command:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998
  ```

- **REDBOOKS Category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.html). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

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** — 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 Publications Customer Support P.O. Box 29570 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark

- **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)001-408-256-5422 (Outside USA)** — ask for:

Index # 4421 Abstracts of new redbooks
Index # 4422 IBM redbooks
Index # 4420 Redbooks for last six months

- **On the World Wide Web**

Redbooks Web Site	http://www.redbooks.ibm.com/
IBM Direct Publications Catalog	http://www.elink.ibmlink.ibm.com/pbl/pbl

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (`http://www.redbooks.ibm.com/redpieces.html`). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number
- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

ITSO Redbook Evaluation

OS/390 Release 5 Implementation
SG24-5151-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. **Please complete this questionnaire and return it using one of the following methods:**

- Use the online evaluation form found at http://www.redbooks.ibm.com
- Fax this form to: USA International Access Code 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
__Customer __Business Partner __Solution Developer __IBM employee
__None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction _____

Please answer the following questions:

Was this redbook published in time for your needs? Yes____ No____

If no, please explain:

_____

_____

_____

_____

What other redbooks would you like to see published?

_____

_____

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

_____

_____

_____

_____

_____

SG24-5151-00
Printed in the U.S.A.