z/OS Communications Server
2.5

IP Network Print Facility

IBM

SC27-3658-50

Note:

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 167.

This edition applies to Version 2 Release 5 of z/OS® (5650-ZOS), and to subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2021-09-11

Contents

Figures

Tables

About this document

This book describes how to plan for, customize, and use the Network Print Facility (NPF) in Transmission Control Protocol/Internet Protocol (TCP/IP) networks.

This document is for system programmers and network administrators who need to prepare their network to route VTAM®, JES2, or JES3 printer output to remote printers .

You should be familiar with either JES or VTAM or both, to use this document.

You must perform several tasks to benefit from the Network Print Facility. The following table shows where you can find information on each of the tasks required. When information is located in a book other than this one, the title is noted.

Table 1. Network Print Facility implementation Tasks

For This Task:	See This Information in This Book:
Planning for the Network Print Facility	See Chapter 2, “Planning for the Network Print Facility,” on page 11.
Installing the Network Print Facility	Installation appears in the z/OS Program Directory
Configuring the Network Print Facility	See Chapter 3, “Customizing the Network Print Facility,” on page 15.
Learning about the NPF files	See Chapter 4, “Understanding the Network Print Facility files,” on page 37.
Using macros to initialize and load the files	See Chapter 5, “Creating the Network Print Facility files,” on page 43.
Using the Network Print Facility panels to maintain the files and manage the print queues	See Chapter 6, “Maintaining the Network Print Facility files with ISPF,” on page 59.
Writing user exits to customize the Network Print Facility	See Chapter 7, “Writing exit routines to tailor the Network Print Facility,” on page 93.
Starting, operating, and stopping the Network Print Facility	See Chapter 8, “Operating the Network Print Facility,” on page 111.
Diagnosing Network Print Facility problems	See Chapter 9, “Diagnosing Network Print Facility Problems,” on page 125. Messages and codes appear in z/OS Communications Server: IP Messages Volume 1 (EZA).

How to use this document

To use this document, you should be familiar with z/OS TCP/IP Services and the TCP/IP suite of protocols.

How to contact IBM service

For immediate assistance, visit this website: https://www.ibm.com/mysupport

Most problems can be resolved at this website, where you can submit questions and problem reports electronically, and access a variety of diagnosis information.

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-IBM®-SERV). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

If you would like to provide feedback on this publication, see "Communicating your comments to IBM" on page 181.

Conventions and terminology that are used in this information

Commands in this information that can be used in both TSO and z/OS UNIX environments use the following conventions:

- When describing how to use the command in a TSO environment, the command is presented in uppercase (for example, NETSTAT).
- When describing how to use the command in a z/OS UNIX environment, the command is presented in bold lowercase (for example, **netstat**).
- When referring to the command in a general way in text, the command is presented with an initial capital letter (for example, Netstat).

All the exit routines described in this information are *installation-wide exit routines*. The installation-wide exit routines also called installation-wide exits, exit routines, and exits throughout this information.

The TPF logon manager, although included with VTAM, is an application program; therefore, the logon manager is documented separately from VTAM.

Samples used in this information might not be updated for each release. Evaluate a sample carefully before applying it to your system.

z/OS no longer supports mounting HFS data sets (The POSIX style file system). Instead, a z/OS File System (ZFS) can be implemented. The term hierarchical file system, abbreviated as HFS, is defined as a data structure that has a hierarchical nature with directories and files. References to hierarchical file systems or HFS might still be in use in z/OS Communications Server publications.

Note: In this information, you might see the following Shared Memory Communications over Remote Direct Memory Access (SMC-R) terminology:

- RoCE Express®, which is a generic term representing IBM 10 GbE RoCE Express, IBM 10 GbE RoCE Express2, and IBM 25 GbE RoCE Express2 feature capabilities. When this term is used in this information, the processing being described applies to all of these features. If processing is applicable to only one feature, the full terminology, for instance, IBM 10 GbE RoCE Express will be used.
- RoCE Express2, which is a generic term representing an IBM RoCE Express2® feature that might operate in either 10 GbE or 25 GbE link speed. When this term is used in this information, the processing being described applies to either link speed. If processing is applicable to only one link speed, the full terminology, for instance, IBM 25 GbE RoCE Express2 will be used.
- RDMA network interface card (RNIC), which is used to refer to the IBM 10 GbE RoCE Express, IBM® 10 GbE RoCE Express2, or IBM 25 GbE RoCE Express2 feature.
- Shared RoCE environment, which means that the "RoCE Express" feature can be used concurrently, or shared, by multiple operating system instances. The feature is considered to operate in a shared RoCE environment even if you use it with a single operating system instance.

Clarification of notes

Information traditionally qualified as Notes is further qualified as follows:

Attention
Indicate the possibility of damage

Guideline
Customary way to perform a procedure

Note
Supplemental detail

Rule

Something you must do; limitations on your actions

Restriction

Indicates certain conditions are not supported; limitations on a product or facility

Requirement

Dependencies, prerequisites

Result

Indicates the outcome

Tip

Offers shortcuts or alternative ways of performing an action; a hint

How to read a syntax diagram

This syntax information applies to all commands and statements that do not have their own syntax described elsewhere.

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line (the main path).

Symbols and punctuation

The following symbols are used in syntax diagrams:

Symbol

Description

►►

Marks the beginning of the command syntax.

►

Indicates that the command syntax is continued.

|

Marks the beginning and end of a fragment or part of the command syntax.

►◄

Marks the end of the command syntax.

You must include all punctuation such as colons, semicolons, commas, quotation marks, and minus signs that are shown in the syntax diagram.

Commands

Commands that can be used in both TSO and z/OS UNIX environments use the following conventions in syntax diagrams:

- When describing how to use the command in a TSO environment, the command is presented in uppercase (for example, NETSTAT).
- When describing how to use the command in a z/OS UNIX environment, the command is presented in bold lowercase (for example, netstat).

Parameters

The following types of parameters are used in syntax diagrams.

Required

Required parameters are displayed on the main path.

Optional

Optional parameters are displayed below the main path.

Default

Default parameters are displayed above the main path.

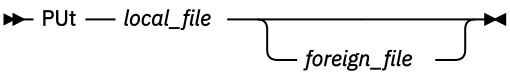
Parameters are classified as keywords or variables. For the TSO and MVS™ console commands, the keywords are not case sensitive. You can code them in uppercase or lowercase. If the keyword appears in the syntax diagram in both uppercase and lowercase, the uppercase portion is the abbreviation for the keyword (for example, OPERand).

For the z/OS UNIX commands, the keywords must be entered in the case indicated in the syntax diagram.

Variables are italicized, appear in lowercase letters, and represent names or values you supply. For example, a data set is a variable.

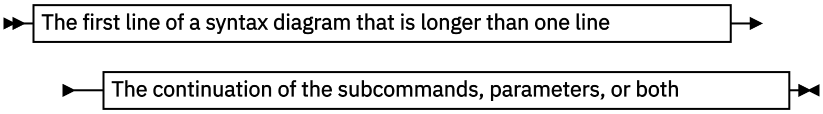
Syntax examples

In the following example, the PUt subcommand is a keyword. The required variable parameter is *local_file*, and the optional variable parameter is *foreign_file*. Replace the variable parameters with your own values.



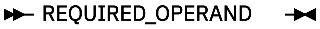
Longer than one line

If a diagram is longer than one line, the first line ends with a single arrowhead and the second line begins with a single arrowhead.


The first line of a syntax diagram that is longer than one line
The continuation of the subcommands, parameters, or both

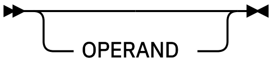
Required operands

Required operands and values appear on the main path line. You must code required operands and values.



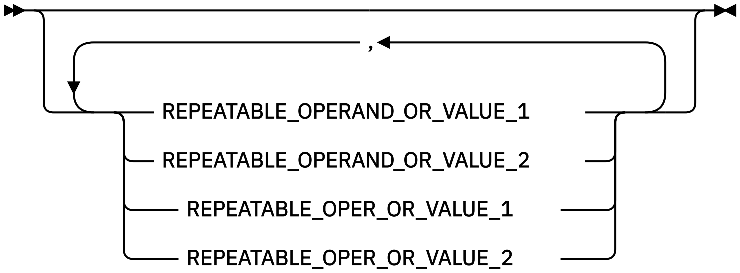
Optional values

Optional operands and values appear below the main path line. You do not have to code optional operands and values.



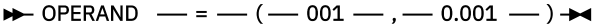
Selecting more than one operand

An arrow returning to the left above a group of operands or values means more than one can be selected, or a single one can be repeated.


,
REPEATABLE_OPERAND_OR_VALUE_1
REPEATABLE_OPERAND_OR_VALUE_2
REPEATABLE_OPER_OR_VALUE_1
REPEATABLE_OPER_OR_VALUE_2

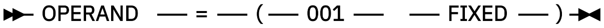
Nonalphanumeric characters

If a diagram shows a character that is not alphanumeric (such as parentheses, periods, commas, and equal signs), you must code the character as part of the syntax. In this example, you must code OPERAND=(001,0.001).



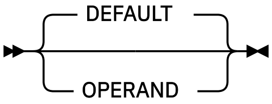
Blank spaces in syntax diagrams

If a diagram shows a blank space, you must code the blank space as part of the syntax. In this example, you must code OPERAND=(001 FIXED).



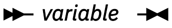
Default operands

Default operands and values appear above the main path line. TCP/IP uses the default if you omit the operand entirely.



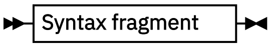
Variables

A word in all lowercase italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

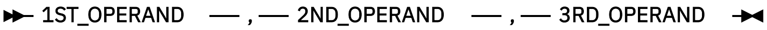


Syntax fragments

Some diagrams contain syntax fragments, which serve to break up diagrams that are too long, too complex, or too repetitious. Syntax fragment names are in mixed case and are shown in the diagram and in the heading of the fragment. The fragment is placed below the main diagram.



Syntax fragment



Prerequisite and related information

z/OS Communications Server function is described in the z/OS Communications Server library. Descriptions of those documents are listed in "Bibliography" on page 171, in the back of this document.

Required information

Before using this product, you should be familiar with TCP/IP, VTAM, MVS, and UNIX System Services.

Softcopy information

Softcopy publications are available in the following collection.

Titles	Description
IBM Z Redbooks	The IBM Z®® subject areas range from e-business application development and enablement to hardware, networking, Linux®, solutions, security, parallel sysplex, and many others. For more information about the Redbooks® publications, see http://www.redbooks.ibm.com/ and http://www.ibm.com/systems/z/os/zos/zfavorites/.

Other documents

This information explains how z/OS references information in other documents.

When possible, this information uses cross-document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see z/OS Information Roadmap (SA23-2299). The Roadmap describes what level of documents are supplied with each release of z/OS Communications Server, and also describes each z/OS publication.

To find the complete z/OS library, visit the z/OS library in IBM Documentation (https://www.ibm.com/docs/en/zos).

Relevant RFCs are listed in an appendix of the IP documents. Architectural specifications for the SNA protocol are listed in an appendix of the SNA documents.

The following table lists documents that might be helpful to readers.

Title	Number
DNS and BIND, Fifth Edition, O'Reilly Media, 2006	ISBN 13: 978-0596100575
Routing in the Internet, Second Edition, Christian Huitema (Prentice Hall 1999)	ISBN 13: 978-0130226471
sendmail, Fourth Edition, Bryan Costales, Claus Assmann, George Jansen, and Gregory Shapiro, O'Reilly Media, 2007	ISBN 13: 978-0596510299
SNA Formats	GA27-3136
TCP/IP Illustrated, Volume 1: The Protocols, W. Richard Stevens, Addison-Wesley Professional, 1994	ISBN 13: 978-0201633467
TCP/IP Illustrated, Volume 2: The Implementation, Gary R. Wright and W. Richard Stevens, Addison-Wesley Professional, 1995	ISBN 13: 978-0201633542
TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, W. Richard Stevens, Addison-Wesley Professional, 1996	ISBN 13: 978-0201634952
TCP/IP Tutorial and Technical Overview	GG24-3376
Understanding LDAP	SG24-4986
z/OS Cryptographic Services System SSL Programming	*SC14-7495*

Title	Number
z/OS IBM Tivoli Directory Server Administration and Use for z/OS	*SC23-6788*
z/OS JES2 Initialization and Tuning Guide	*SA32-0991*
z/OS Problem Management	*SC23-6844*
z/OS MVS Diagnosis: Reference	*GA32-0904*
z/OS MVS Diagnosis: Tools and Service Aids	*GA32-0905*
z/OS MVS Using the Subsystem Interface	*SA38-0679*
z/OS Program Directory	*GI11-9848*
z/OS UNIX System Services Command Reference	*SA23-2280*
z/OS UNIX System Services Planning	*GA32-0884*
z/OS UNIX System Services Programming: Assembler Callable Services Reference	*SA23-2281*
z/OS UNIX System Services User's Guide	*SA23-2279*
z/OS XL C/C++ Runtime Library Reference	*SC14-7314*
Open Systems Adapter-Express Customer's Guide and Reference	*SA22-7935*

Redbooks publications

The following Redbooks publications might help you as you implement z/OS Communications Server.

Title	Number
IBM z/OS Communications Server TCP/IP Implementation, Volume 1: Base Functions, Connectivity, and Routing	SG24-8096
IBM z/OS Communications Server TCP/IP Implementation, Volume 2: Standard Applications	SG24-8097
IBM z/OS Communications Server TCP/IP Implementation, Volume 3: High Availability, Scalability, and Performance	SG24-8098
IBM z/OS Communications Server TCP/IP Implementation, Volume 4: Security and Policy-Based Networking	SG24-8099
IBM Communication Controller Migration Guide	SG24-6298
IP Network Design Guide	SG24-2580
Managing OS/390 TCP/IP with SNMP	SG24-5866
Migrating Subarea Networks to an IP Infrastructure Using Enterprise Extender	SG24-5957
SecureWay Communications Server for OS/390 V2R8 TCP/IP: Guide to Enhancements	SG24-5631
SNA and TCP/IP Integration	SG24-5291
TCP/IP in a Sysplex	SG24-5235
TCP/IP Tutorial and Technical Overview	GG24-3376
Threadsafe Considerations for CICS	SG24-6351

Where to find related information on the Internet

z/OS

This site provides information about z/OS Communications Server release availability, migration information, downloads, and links to information about z/OS technology

http://www.ibm.com/systems/z/os/zos/

z/OS Internet Library

Use this site to view and download z/OS Communications Server documentation

http://www.ibm.com/systems/z/os/zos/library/bkserv/

z/OS Communications Server product

The page contains z/OS Communications Server product introduction

https://www.ibm.com/products/zos-communications-server

IBM Communications Server product support

Use this site to submit and track problems and search the z/OS Communications Server knowledge base for Technotes, FAQs, white papers, and other z/OS Communications Server information

https://www.ibm.com/mysupport

IBM Communications Server performance information

This site contains links to the most recent Communications Server performance reports

http://www.ibm.com/support/docview.wss?uid=swg27005524

IBM Systems Center publications

Use this site to view and order Redbooks publications, Redpapers, and Technotes

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

z/OS Support Community

Search the z/OS Support Community Library for Techdocs (including Flashes, presentations, Technotes, FAQs, white papers, Customer Support Plans, and Skills Transfer information)

z/OS Support Community

Tivoli® NetView® for z/OS

Use this site to view and download product documentation about Tivoli NetView for z/OS

http://www.ibm.com/support/knowledgecenter/SSZJDU/welcome

RFCs

Search for and view Request for Comments documents in this section of the Internet Engineering Task Force website, with links to the RFC repository and the IETF Working Groups web page

http://www.ietf.org/rfc.html

Internet drafts

View Internet-Drafts, which are working documents of the Internet Engineering Task Force (IETF) and other groups, in this section of the Internet Engineering Task Force website

http://www.ietf.org/ID.html

Information about web addresses can also be found in information APAR II11334.

Note: Any pointers in this publication to websites are provided for convenience only and do not serve as an endorsement of these websites.

DNS websites

For more information about DNS, see the following USENET news groups and mailing addresses:

USENET news groups
comp.protocols.dns.bind

BIND mailing lists
https://lists.isc.org/mailman/listinfo

BIND Users

- Subscribe by sending mail to bind-users-request@isc.org.
- Submit questions or answers to this forum by sending mail to bind-users@isc.org.

BIND 9 Users (This list might not be maintained indefinitely.)

- Subscribe by sending mail to bind9-users-request@isc.org.
- Submit questions or answers to this forum by sending mail to bind9-users@isc.org.

The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of Information Technology professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS systems programmer.

Specifically, the z/OS Basic Skills Information Center is intended to achieve the following objectives:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS

To access the z/OS Basic Skills Information Center, open your web browser to the following website, which is available to all users (no login required): https://www.ibm.com/support/knowledgecenter/zosbasics/com.ibm.zos.zbasics/homepage.html?cp=zosbasics

Summary of changes

This document contains terminology, maintenance, and editorial changes, including changes to improve consistency and retrievability. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

Changes made in z/OS Communications Server Version 2 Release 5

This information contains no technical changes for this release.

Changes made in z/OS Communications Server Version 2 Release 4

This information contains no technical changes for this release.

Changes made in z/OS Communications Server Version 2 Release 3

This information contains no technical changes for this release.

Chapter 1. Introduction

The Network Print Facility (NPF) is a separately orderable feature of z/OS Communications Server. The Network Print Facility routes JES or VTAM print data from your MVS system to printers in the TCP/IP network.

Software requirements

In addition to the base operating system requirements for TCP/IP for V3R2, Network Print Facility requires software for certain environments.

To do any printing that requires a conversion of host names to IP addresses, the Network Print Facility requires:

- IBM C/370 V2.2, or LE/370 V1.3 or higher, C run-time libraries

To process VTAM print data, the Network Print Facility requires VTAM 3.4.2 or higher.

To process JES print data, the Network Print Facility requires:

- JES2 SP Version 4 or higher
- JES3 4.2.1 or higher with APAR OY50274

To process JES print data with LRECL greater than 4088 bytes, the Network Print Facility requires:

- JES2 V4 or later with APARs OW07587 and OW07588
- JES3 V4.2.1 or later with APARs OW07588 and OW07589

To use the NPF ISPF panel interface, the Network Print Facility requires:

- ISPF 3.5 or higher
- IBM C/370 V2.2, or LE/370 V1.3 or higher, C run-time libraries

Overview

The Network Print Facility lets you print data from your z/OS system on remote printers accessible through IBM IP Services. The Network Print Facility supports the printing of the following types of output:

- JES2 output
- JES3 output
- VTAM SNA character string (SCS) output over LU type 1 sessions
- VTAM 3270 data stream output over LU type 3 and LU type 0 sessions

The Network Print Facility accomplishes this by transforming VTAM or JES output print data into a format that existing LPD functions can process.

The Network Print Facility lets you decide ***where*** and ***how*** output will be printed through the use of a routing file and options file. The ***where*** portion is defined by the routing file. The ***how*** portion is defined through LPR options in the options file and through other data in the routing file.

The Network Print Facility allows user exits for installation-defined routing decisions or data modifications. It also provides a queue manager program to control the initial sending of each print job, retries of failed print jobs, and deletion of print data sets after a user-specified retention time.

The Network Print Facility incorporates the following functions:

- For communication with printers in the TCP/IP network, the Network Print Facility uses the protocol defined in Request for Comment (RFC) 1179 and amendments.

  - The Line Printer Daemon (LPD) is the remote print server defined by this protocol. The LPD can have queueing capability, in which case it performs as a print spooling mechanism and a print driver.

    In this book, the term LPD refers to any print server which observes this protocol; it is not meant to imply any specific product implementation.
  - The Line Printer Requestor (LPR) is the client defined by this protocol. LPR code within the Network Print Facility manages the communication between the MVS system and the LPD.

    The Network Print Facility supports the same options as TCP/IP's LPR command, including translation. For more detail, see the LPR options described in the z/OS Communications Server: IP User's Guide and Commands and the translation tables topic in z/OS Communications Server: IP Configuration Guide.
- Two capture points allow the printing of data from either JES or VTAM:
  - For JES, an FSS writer for which the network administrator must specify one or more output classes. Output queued under any of those specified classes is routed by JES to the NPF FSS writer, which then prepares it for transmission to the LPD server.
  - For VTAM, a VTAM capture point application program, which does processing for one or more logical printers. Each logical printer emulates a real SNA-network printer, supporting SNA character string (SCS) data streams over an LU type-1 session or 3270 data streams over an LU type-3 or LU type-0 session. For more information about SCS and 3270 data streams, see *SNA - Sessions Between Logical Units* and the *3270 Information Display System Data Stream Programmer's Reference*.

    An application with data to print (usually CICS® or IMS) must first start a session with a logical printer in the NPF VTAM capture point application. When print data is received over that session, NPF then prepares it for transmission to the LPD server.

    On the session, the NPF logical printer always acts as the secondary LU (SLU), and the application sending the print data acts as the primary LU (PLU). In this book, the sender of the print data will often be referred to as the **session partner LU.**

- A combination of internet name or IP address and printer name allow routing of print data within the TCP/IP network. Because this information is not provided by MVS applications that use JES or VTAM to print data, the Network Print Facility provides a routing mechanism. The routing mechanism uses a combination of files and optional user exits to derive the internet name or IP address and printer name from data provided by MVS.

  The internet routings and print functions are determined in either of 2 ways or by a combination of the 2:

  1. Routing and options files created by the network administrator.

     These files provide information for routing and printing based on the following subsystems:

     VTAM
       logical printer name

     JES
       Class, destination, and forms (JCL parameters CLASS, DEST, FORMS)

     This information is used to build the key used to access the routing file. (These keys are referred to as major and minor names. Definitions for these names are within the parameter descriptions in "EZAPPFL TYPE=ROUTING" on page 45.) The routing file in turn contains the key of a record in the options file. Between the 2 records, the system constructs a routing data area, which contains both routing information and printing options. This routing data area is used to determine the actual printer destination.
  2. User exits

     The Network Print Facility provides for user-written exits that can either replace the functions of the routing and options files or augment the information found therein.

  - The optional general routing exit (user-supplied), invoked prior to the retrieval of the routing record from the routing file, can replace the routing file function by creating the routing data area, or it can alter the key used to access the routing file.
  - The optional specific routing exit (user-supplied), invoked after the retrieval of the routing record or the invocation of the general routing exit, can alter the routing data area, including the internet name or IP address and the printer name.
  - The optional input record exit (user-supplied), invoked as each print record is processed to modify the print data stream (its principal purpose), can also modify the routing data area based on information found within the print stream.

  Each user exit is described in Chapter 7, "Writing exit routines to tailor the Network Print Facility," on page 93.

- The Network Print Facility's file creation macro and an ISPF interface can be used to build the routing and options files. See Chapter 5, "Creating the Network Print Facility files," on page 43 and Chapter 6, "Maintaining the Network Print Facility files with ISPF," on page 59 for more detail.
- For each individual print job, the Network Print Facility creates a sequential print data set and adds a corresponding record to its queue file. The queue file record keeps track of the print data set's name and status, along with the processing rules copied from the routing data area. The Network Print Facility's queue manager periodically scans the queue file and performs actions such as initial send attempts, retries, retention or deletion on files as required. The ISPF panel interface can be used to display or modify queue file record data.
- The Network Print Facility normally uses multiple address spaces.

  For each of the following types of applications within NPF, there can be 0, 1, or multiple instances of the application running at any given time, and each instance of the application is a separately started MVS job or procedure running in its own address space:

  - The VTAM capture point application
  - The FSS writer acting as the JES capture point
  - The queue manager

  In addition, a TSO/E user authorized to use the Network Print Facility ISPF interface will require his own address space.

Flow of Network Print Facility Processing

Network Print Facility processing is illustrated in Figure 1 on page 4, which contains numbered boxes that correlate to the numbered items in the list below.

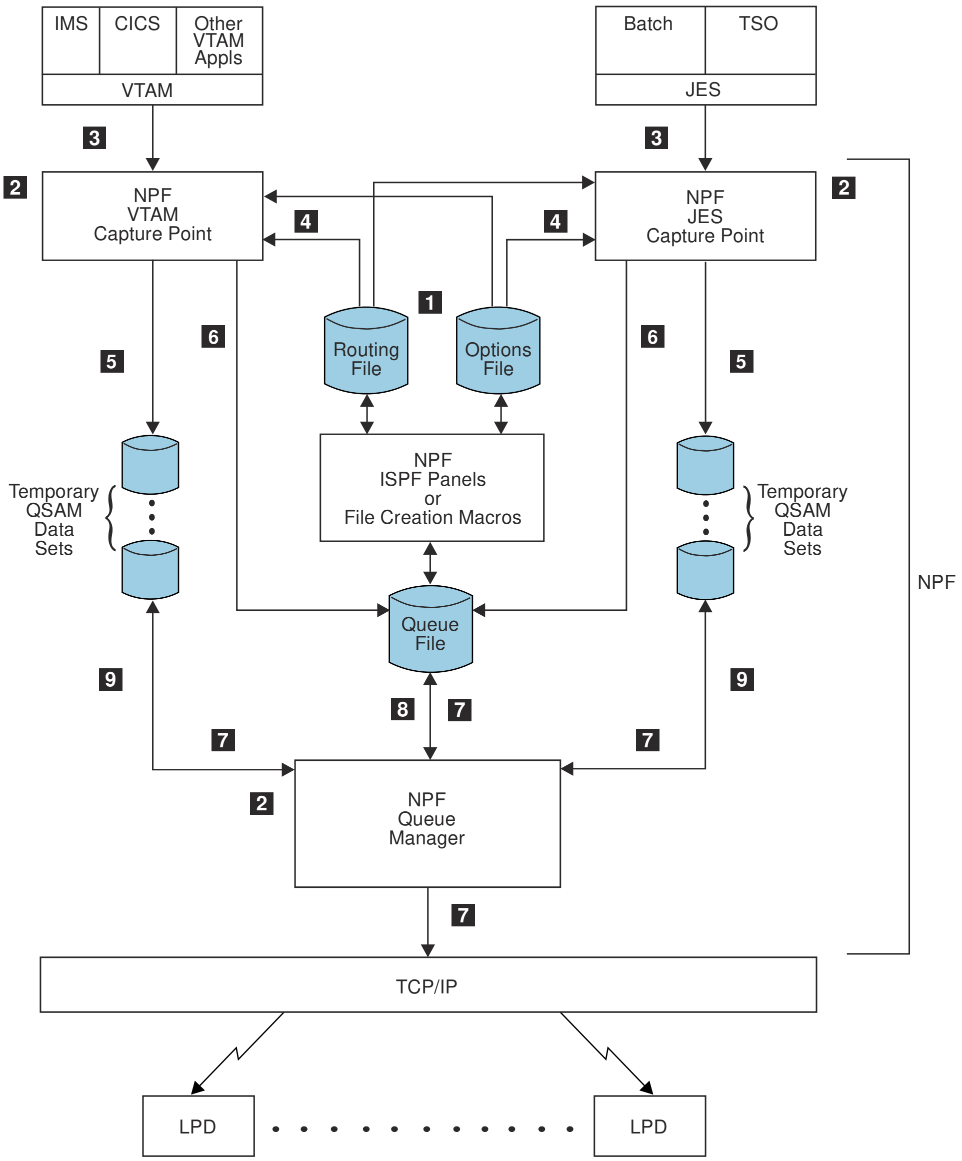


Figure 1. Flow of Network Print Facility processing

The system administrator creates routing and options files using the macros provided by Network Print Facility. These files can be modified through the use of the ISPF interface, after the initial loading. The macros and the ISPF panel interface, or both, can also be used to add data to the files.

The operator starts the Network Print Facility JES or VTAM capture point application and starts the queue manager.

For each print job, the Network Print Facility receives output through either the JES or VTAM capture point application.

The Network Print Facility obtains the necessary routing and options file information for the output. The routing can be to a single printer (a normal routing) or to multiple printers of the same type (a specific broadcast routing). Users can supply general routing or specific routing exits to alter the routing of print output.

The Network Print Facility creates a sequential print data set from the output received. Users can supply an input record exit to alter the print output before it is queued in this data set.

The Network Print Facility adds a record to the queue file for each location at which the print data set is to be printed. The queue file record contains information from the routing and options files in addition to the name and status of the print data set.

If multiple print jobs are waiting to go to the same printer, the NPF queue manager processes them in FIFO (First In, First Out) order. When it is time to send a particular print job, the queue manager uses information from the queue file to forward the queued data set to LPD services for printing.

Based on the return code from LPD and information in the queue file record, the NPF queue manager handles the queue record in one of the following ways:

- Keeps it to force the retry of an unsuccessful transmission.
- Keeps it for a user-specified retain time after a successful transmission or after all retries have been exhausted for an unsuccessful transmission.
- Erases it if no retry or retain time is indicated.

The NPF queue manager deletes the print data set after the last queue file record referring to it has been erased. For more details about queue manager processing, see "Managing the print queues" on page 85.

Print Data Processing

Within NPF, print data can be transformed at three different points in the flow. These transformations occur in the following order:

- VTAM only - The SCS or 3270 data stream received from the session partner LU is converted into a series of variable-length print records, each starting with an ASA carriage-control character.
- JES or VTAM - If the user has provided an input record exit, that exit is given the opportunity to modify each print record before it is queued.
- JES or VTAM - Prior to transmission to LPD, the print data set can be transformed as directed by user-specified LPR options. The same transformations available via the stand-alone LPR command can be used within the Network Print Facility, including the following information:
  - Translation
  - PostScript** output
  - Page header support

Network Print Facility interfaces with JES

An application generating print data is designed for a specific print environment, so an MVS application printing through JES spools might expect to create a print file having 132 alphanumeric character line width using ASA control characters for forms control. Or, it might create a print file a page at a time using graphic symbols. The application is generally unaware of JES or its functions, at least from a programming perspective. The person who defines the run-time environment for applications producing print for JES distribution is more aware of JES specifying such parameters as CLASS, FORMS, and DEST in the JCL.

The Network Print Facility capture point for JES appears to the application as a printer through the use of a special FSS writer as a part of the Network Print Facility. Certain definitions will aid further explanation:

FSI
The Functional Subsystem Interface (FSI). This MVS interface allows communication between JES and your functional subsystem and functional subsystem application. This interface is what the Network Print Facility uses to communicate with MVS and JES to process JES output.

FSS
The Functional Subsystem. This MVS subsystem is a collection of programs in an address space separate from JES that provides a JES-related function, such as print processing. An FSS extends the scope of JES processing.

Writer
A JES function that processes print output.

NPF FSS writer
The special writer, part of the Network Print Facility, that you define to capture JES output. Each runs in a separate address space.

FSA

A functional subsystem application is a collection of programs in the FSS address space that control one device.

NPF FSA subtask

The FSA subtask that the Network Print Facility uses to control each logical printer. One or more of these can run under each NPF FSS writer you establish.

NPF JES logical printer

There is a one-to-one correlation between the NPF JES logical printer and each FSA subtask.

LPD print queue

The LPD print queue on a host system that prints the output that has been transferred through each of the above.

This NPF FSS writer interfaces with JES in the following ways:

- At NPF FSS writer startup and initialization
- Accepting notification from JES that there is a print spool data set available for processing
- Using the MVS FSI to retrieve the print spool data set from the JES queue.
- Processing the data set; sending it forward from the FSA subtask through the NPF JES logical printer and eventually to a QSAM print data set pointed to by a queue file record.

Figure 2 on page 6 illustrates the relationship of the NPF FSS writer, the FSA subtask, and the logical printers.

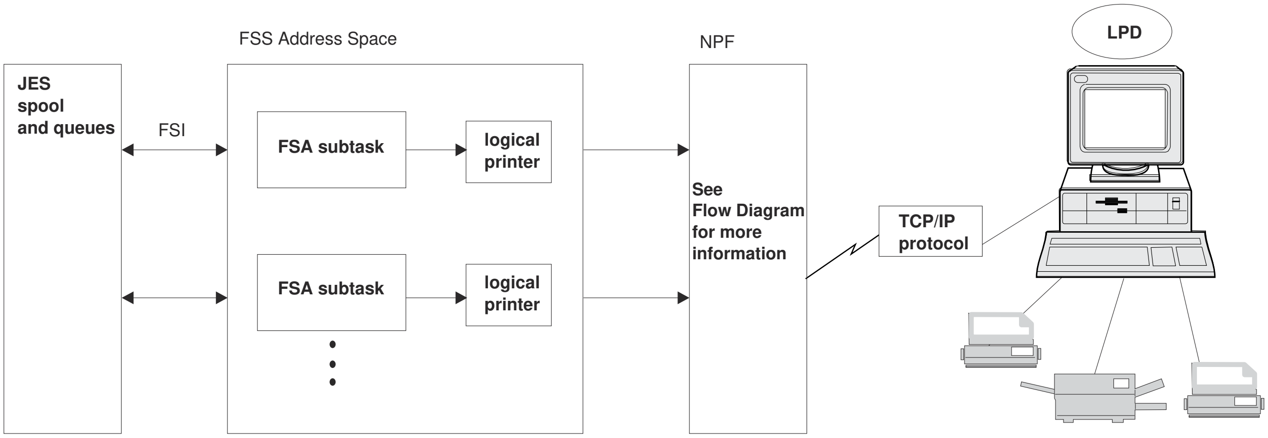


Figure 2. NPF JES elements

As illustrated in Figure 2 on page 6, the Network Print Facility FSS writer uses the Functional Subsystem Interface (FSI) to communicate with MVS and JES. It communicates with MVS JES to receive output data sets (SYSOUT) from JES spool and write the data into a file for transmission to LPD.

Network Print Facility interfaces with VTAM

To do printing, a VTAM application program typically starts a session with a printer in its SNA network and sends the print data during that session. The Network Print Facility replaces the SNA-network printer with a logical printer within an NPF VTAM capture point application program.

For each logical printer LU:

- VTAM must activate an APPL definition statement
- The NPF VTAM capture point application must open a matching ACB

The NPF VTAM capture point application as a whole is not defined to VTAM in any way.

It is possible to run multiple instances of the NPF VTAM capture point application at the same time. Each must run in a different MVS address space and must do processing for a different set of logical printer LUs. There are several reasons why this might be useful:

- To improve throughput by spreading processing across multiple address spaces.
- To reduce region size problems by spreading the logical printer LUs across multiple regions.
- To enable different sets of logical printer LUs to be started and stopped at different times.

The set of logical printer LUs to be supported by any given instance of the NPF VTAM capture point application is selected as follows:

- The routing file entry for each logical printer LU contains a class identifier, the LUCLASS parameter, which specifies one or more numbers in the range from 1 to 64. LUCLASS is specified via the ROUTING function of the EZAPPFL macro. See "EZAPPFL TYPE=ROUTING" on page 45 for a complete definition of the LUCLASS parameter.
- The JCL used to start each instance of the NPF VTAM capture point application contains an LUCLASS sub-parameter, also specifying one or more numbers in the range from 1 to 64.
- Each logical printer LU whose routing file record's LUCLASS value matches the JCL's LUCLASS value is selected. The LUCLASS values are considered to match if they have one or more class identifiers in common; it is not necessary to have all specified class identifiers the same.

When running the VTAM capture point application, the routing file is required for logical printer name selection, even if the coding of a general routing exit makes it unnecessary for actual routing functions.

Specific concepts for processing VTAM output

This section describes some special topics pertinent to VTAM users.

BIND parameters

For the BIND sent to start a session with a Network Print Facility logical printer, there are some parameter restrictions. The LU Type parameter must be consistent with the type of print data to be sent during the session, and the FM Profile and TS Profile values must be allowed for that LU Type. Valid combinations of these BIND parameters are shown below:

Table 2. VTAM BIND parameter requirements

Data Type	LU Type	FM Profile	TS Profile
3270 data	LU 0	2	2
3270 data	LU 3	3	3
SNA character string (SCS) data	LU 1	3 or 4	3 or 4

Each BIND specifying an LU type of 0 or 3 must also specify a valid non-zero screen size.

In VTAM, various sets of default BIND parameters are defined by entries in user-modifiable logmode tables. Either explicitly or by default, the APPL definition for each Network Print Facility logical printer is associated with one of these logmode table entries containing a specific set of BIND parameters. When starting a session, the PLU uses these default BIND parameters, or it uses its own resource definitions to override some or all of the default BIND parameter data.

Attention: When an IMS application is the PLU and the logical printer is defined in IMS as a non-SNA 3270 device, IMS uses the default BIND parameters without any changes. Therefore, those default parameters must be correct for a 3270 data stream as illustrated in Table 2 on page 7.

In most other cases, CICS and IMS subsystems override default BIND parameters as necessary to ensure consistency. With session partner LUs other than CICS or IMS, make sure that the combination of default BIND parameters in the logmode table and parameter overrides by the PLU result in a correct BIND for the Network Print Facility.

Restrictions on data stream support

For LU 1 sessions, Network Print Facility does not support:

- FM headers - For LU 1 data streams, the Network Print Facility VTAM capture point application does not expect FM data requests that it receives to be formatted. It has no logic to handle FM headers. Any request containing an FM header is rejected with a sense code of X'10030000' (function not supported).
- Some SCS commands. See Appendix D, “The SCS printer emulator,” on page 141 for more detail.

For LU 3 or LU 0 sessions, Network Print Facility does not support:

- Anything other than the basic function set for the 3270 data stream as described in *SNA - Sessions Between Logical Units*. Note that the Write Structured Fields command is not included in the basic function set and, therefore, is not supported.
- DBCS

Differences between SNA and TCP/IP print processing

TCP/IP-attached printers accessed by the Network Print Facility do not behave exactly like their SNA-network counterparts.

- The SNA-network user receives verification that the data has been printed:
  - In SNA, a positive response from a printer means that the data has actually been printed successfully.
  - With the Network Print Facility, a positive response means that the data has been received successfully by the Network Print Facility VTAM capture point application and queued successfully in a QSAM data set.
- If encryption is being used, the Network Print Facility only protects the data's security across the SNA LU-LU session:
  - With a real SNA-network printer, the encryption protects the data all the way to the final destination.
  - With the Network Print Facility, the data is decrypted as it arrives in the Network Print Facility application's VTAM host; it is unprotected across the TCP/IP portion of the network.
- If compression is being used, VTAM-supported data compression is available only between the sender and Network Print Facility's VTAM host.
  - With a real SNA-network printer, the compression is available all the way to the final destination.
  - With the Network Print Facility, the compressed data is decompressed after it arrives in the Network Print Facility application's VTAM host, but before it arrives at the application itself, not across the TCP/IP portion of the network.
- In an SNA network, being in session with a printer gives the application exclusive control of that printer, except for possible interference from local copy operations. The printer's normal session limit of 1 prevents any other application's print output from being interleaved with the owning application's output.

  In a TCP/IP network, such exclusive control of the printer is not generally possible because a session limit of 1 applies only to the SNA portion of the flow, and it will not stop other users in the TCP/IP network from sending print jobs to the real printer that the logical printer represents.

User-modifiable tables

NPF's VTAM capture point application uses two user-modifiable tables to define certain processing rules for its sessions. These tables are introduced here and more fully explained later in the book.

- The end-of-file rules table

  The Network Print Facility uses the concept of a print **file**, defined as:

  > A batch of related print data which should be handled as a single document. NPF keeps all of the data for a file together as it is queued, transmitted to LPD, and printed.

For correct results, it is necessary for NPF's VTAM capture point application to determine accurately where one file should end and the next one should begin. Unfortunately, it is not always clear how to do this.

To help in making these decisions, the Network Print Facility provides a user-modifiable end-of-file rules table. This table allows different end-of-file rules to be defined for different combinations of logical printer LU, session partner LU, and LU type used on the session. See "File definition for VTAM output" on page 25 for more information about how the Network Print Facility defines file boundaries.

- The default page format table

  In some cases, the SCS data streams used on LU type 1 sessions depend on a default page format which is hard-wired or otherwise pre-set for the printer being used.

  The Network Print Facility provides a user-modifiable table to allow the equivalent of these hard-wired defaults to be defined for its logical printers. See "Page format definition" on page 32 for more details on the Network Print Facility's page formats.

Notes for IMS/CICS users of telnet

In IMS or CICS subsystems, applications receive communication services from terminal management functions of subsystems. Thus, the IMS programmer thinks of a device as an LTERM and the CICS programmer thinks of a device as a TCTTE entry.

Many IMS and CICS applications have algorithms that derive LTERM or TCTTE names for printers by performing some hashing technique on the input LTERM or TCTTE names. If the Network Print Facility is used with Telnet, this creates problems because Telnet assigns the input LU name (and thereby the LTERM or TCTTE name) arbitrarily from a pool of LU names. If this type of algorithm is in use in your installation and you are using Telnet support, IBM recommends that the IP-LU mapping feature of Telnet be used. This feature allows the specification of the input LU name rather than having it randomly selected from a pool. This would, in turn, permit the application to select the LU name for the printer. See the LUMAP statement described in z/OS Communications Server: IP Configuration Guide.

Chapter 2. Planning for the Network Print Facility

Careful planning for the Network Print Facility is essential. Read all the information concerning the steps to implement this function before turning to the customization task.

Required skills

Before using the information in this section, make sure you have an understanding of JES or VTAM, or both.

Required environment

Before using the information in this section, make sure you have established the correct LPR/LPD environment. If you are connecting to MVS host printers, read about LPD in "Configuring the Remote Printer Server (LPD)" in the z/OS Communications Server: IP Configuration Guide.

See Chapter 1, “Introduction,” on page 1, the *z/OS Program Directory*, or z/OS Planning for Installation for the correct level of supported products.

Overall planning activities

To run the Network Print Facility, you need to plan in the following ways:

- Decide how many printers you will use.
- Ensure LPR/LPD options are established for how output should be printed.

  What LPD options does each printer support? What LPR options should the Network Print Facility process when sending a job to the printer?
- Decide who should have the authority to interactively access the routing, options, and queue files. Who should be able to change destinations and print options? Who should be able to look at the status of print jobs and who should be able to change or delete them? By default, all interactive users can update the routing, options, and queue files. If you have the Resource Access Control Facility (RACF®), you can use it to prevent access or allow read-only or read-write access for the appropriate users.
- Decide how you can effectively implement the routes you require.

  Should jobs be sent to one printer or to multiple homogeneous printers? Homogeneous printers are printers with identical LPR options. These can be similar printers with different locations or destinations. For more information about homogeneous printers, see “The options file” on page 37 and “The routing file” on page 38.
- Assign major and minor names for routing records. Major and minor names are the keys that the Network Print Facility uses to route print output. See the parameter descriptions in “EZAPPFL TYPE=ROUTING” on page 45 for valid specifications.
- Decide how to handle host name resolution. The routing records can specify an internet name or an IP address as the destination. If an internet name is specified, this name will have to be resolved by the Domain Name Server each time an output file is sent.
- Decide if you want to include an option to access the Network Print Facility on the ISPF Primary Option Menu
- Decide whether you will be using the NPF FSS writers or the VTAM capture point, or both.
- Define the Network Print Facility queue management function:
  - Decide how to handle output that ***does*** print successfully. Do you want to delete the job immediately? Do you want to retain the job for a period of time for future use?

  - Decide how to handle output that ***does not*** print successfully. Do you want to resend the data? If so, how many times do you want to attempt to resend? How often to retry? How long do you want to retain the file after all attempts to resend have failed?
  - Be prepared to specify these criteria in the file creation macros and on the panels.
- Determine if user exits are necessary. If so, create them and plan to specify them appropriately in the macros and on the panels.
- Plan to allocate adequate space for the routing, option, queue, and active VSAM files before using the file creation macros.
- Decide if you will use the various EZAPPFL macro functions to just initialize or to initialize and load the routing and options files.
- Decide whether you will update the routing and options files using the various functions of the EZAPPFL macro or the ISPF interface. The samples provided with the EZAPPFL macro show how to update the routing and options files in batch mode. Chapter 6, "Maintaining the Network Print Facility files with ISPF," on page 59 shows how to update them using the ISPF interface.

See the "Checklist of the NPF tasks for all users (JES or VTAM)" on page 131 for a list of items to check as you begin to use the Network Print Facility.

JES planning considerations

Things to consider as you plan for Network Print Facility use in your JES environment are:

- Decide how many NPF FSS writers you will use. Because each FSA subtask processes output for JES logical printers in parallel, system storage constraints might affect the amount of printing and throughput possible. There might be benefits in specifying 1 FSS writer with multiple FSA subtasks instead of multiple FSS writers each with a single FSA subtask.

  You can parallel process output in two ways:

  1. Using a single writer with multiple logical printers
  2. Defining multiple writers with one printer each

  Writers with multiple printers can be more efficient than having multiple writers with one printer each, because each writer has its own address space. If, however, the printers are of different priority or have a large disparity in job length, one printer might develop a backlog, and in this situation, it might be best to use multiple writers. See the topics on calculating the number of FSA subtasks, defining and managing JES3 resources, and running a printer by using an output FSS in either thez/OS JES2 Initialization and Tuning Guide or the z/OS JES3 Initialization and Tuning Guide
- Decide how you can most effectively use the JCL CLASS parameter. The NPF FSS writer uses the CLASS parameter to decide which job to print. JES supplies the NPF FSS writer with SYSOUT based on a set of installation-defined criteria. Therefore IBM recommends you define your selection criteria as CLASS. Multiple writers can use the same class, or each writer can be assigned to a different class.
- The NPF FSS writer uses fields in the job separator page area (IAZJSPA) such as DEST, CLASS, job name, and user ID in addition to extension areas. Ensure your own changes made to this DSECT (or changes made in JES2 exit 23 or JES3 exit 45, for example) are compatible with the Network Print Facility FSS writer.
- Accommodate the Network Print Facility's use of an 8 alphanumeric character name for destination. If you specify something longer in the DEST parameter of the JCL, you need to truncate that name for Network Print Facility use.
- Accommodate the Network Print Facility's effect on JESNEWS. See "Customizing JESNEWS" on page 21 for more information.
- Determine the best NPF run-time options for your print environment.

  The setting of the SPIN option might affect the print processing or generation of JES separator pages. NPF will process output from JES on either a data set level (SPIN=DATASET) or a group level (SPIN=GROUP).

For JES, an output group is defined as "A set of a job's output data sets that share output characteristics, such as class, destination, and external writer." For example, a job's LOG, JCL, and job messages, although three separate JES data sets, can be considered one output group.

See “Modifying the NPF FSS writer start procedure” on page 112 for more information.

VTAM Planning Considerations

Things to consider as you plan for the Network Print Facility use in your VTAM environment are:

- Establish consistent definition of resources.

  The key to consistent definition in the Network Print Facility is the logical printer name. The system administrator must set up a routing file defining the different logical printer names and their corresponding real TCP/IP resource IDs. For VTAM, each logical printer name must also be marked as belonging to one or more LUCLASSes. See “Defining consistent resources” on page 24 for more information.

- Determine the end-of-file rules for logical printers.

  The Network Print Facility provides a default end-of-file rule for use on all sessions. However, for situations where that default rule is not sufficient, you can define different end-of-file rules for various combinations of SLU name, PLU name and LU type. For more information, see “File definition for VTAM output” on page 25.

- Determine default page formats for logical printers.

  The Network Print Facility gives you the opportunity to define default page formats unique to an environment. For more information, see “Page format definition” on page 32.

Chapter 3. Customizing the Network Print Facility

Before you begin customization, both z/OS Communications Server and the Network Print Facility must be installed according to the instructions provided in the *z/OS Program Directory*.

This chapter provides:

- Overall customization information for both JES and VTAM implementations
- Specifications for NPF configuration statements
- JES customization information, including the definition of the functional subsystem (FSS) writers and printers
- VTAM customization information, including the definition of VTAM resources

Overall customization information

Procedure

Perform the following general procedures to customize the Network Print Facility. Use Appendix A, “Checklists for using the Network Print Facility,” on page 131 to verify that you have covered all the planning and customization tasks.

1. Verify configuration statements in TCPIP.DATA. You can find instructions in this chapter.
2. Specify configuration statements in NPF.DATA. You can find instructions in this chapter.
3. Perform specific tasks to customize for JES or VTAM implementations. You can find instructions in this chapter.
4. Create and initialize the routing, options, and queue files. You can find instructions in Chapter 5, “Creating the Network Print Facility files,” on page 43.
5. If required, write the optional user exits. You can find instructions in Chapter 7, “Writing exit routines to tailor the Network Print Facility,” on page 93.
6. Provide access to the NPF ISPF interface and use its panels to add or change data in the files. You can find instructions in Chapter 6, “Maintaining the Network Print Facility files with ISPF,” on page 59.
7. Ensure the JCL you use to start the NPF capture point applications and the queue manager contains the correct data set names for the NPF files.

Results

After you have configured Network Print Facility and initialized and loaded your files, you can start your Network Print Facility capture point applications. See “Starting the NPF VTAM capture point” on page 117 and “Creating the NPF FSS writer start procedure” on page 111 for details.

Configuring NPF using TCPIP.DATA and NPF.DATA statements

When the NPF capture point applications are started, they use configuration statements in the following data sets to set initialization and run-time values:

Data Set	Configuration Statement
TCPIP.DATA	TCPIPJOBNAME DATASETPREFIX

Data Set	Configuration Statement
NPF.DATA	NPFPRINTPREFIX NPFJESALLOCATION NPFVTAMALLOCATION NPFQMGRTHREAD NPFUNIT

TCPIP.DATA configuration

The TCPIP.DATA data set is used by all TCP/IP client applications and is configured as part of z/OS Communications Server customization. You can find more information about this data set and its statements in z/OS Communications Server: IP Configuration Guide.

TCPIP.DATA statements

NPF uses values in two of the TCPIP.DATA configuration statements:

TCPIPJOBNAME provides the job name of the TCP/IP address space. NPF uses this to determine which TCP/IP program to use. Be sure to specify a job name for the TCP/IP address space in the TCPIPJOBNAME statement unless you want the default (TCPIP) to be used.

DATASETPREFIX provides an optional high-level qualifier that can be used as the *hlq* variable when searching for data sets. If NPF cannot find a DATASETPREFIX statement as it goes through the TCPIP.DATA search order, it uses the system default, TCPIP. This *hlq* value can affect the TCPIP.DATA search order, the NPF.DATA search order, and the high-level qualifier used to name NPF's temporary QSAM print data sets.

TCPIP.DATA search order

To find TCPIP.DATA, NPF follows a standard search sequence. It reads each instance of TCPIP.DATA it can find, in order, until it finds values for both the TCP/IP configuration statements. It uses the first value that it finds for each statement. If an allocation fails, the data set does not exist, or the data set is not available, NPF goes to the next data set in the sequence. The search ends when it finds values for both the TCP/IP configuration statements or when all data sets have been checked. The search order is:

1. The data set pointed to by the DD statement //SYSTCPD.
2. A data set with the name of *jobname*.TCPIP.DATA, where *jobname* is the jobname for the NPF capture point application or the TSO user ID for the TSO session (panels).
3. A data set with the name of SYS1.TCPPARMS(TCPDATA).
4. A data set with the name of *hlq*.TCPIP.DATA, where *hlq* is the value set by a DATASETPREFIX statement found in a prior TCPIP.DATA, or the system default value of TCPIP.

You can also find information about the native MVS environment TCPIP.DATA search order, and the system default in z/OS Communications Server: IP Configuration Guide.

NPF.DATA configuration

The NPF.DATA data set is specific for the Network Print Facility feature. A sample is provided in *hlq*.SEZAINST(NPFDATA). It contains configuration statements that define a high-level qualifier and run-time values for both JES and VTAM NPF implementations.

NPF.DATA statements

All of the NPF.DATA configuration statements are optional. If you do not specify them, the Network Print Facility uses the defaults. You can split the NPF.DATA configuration statements across multiple NPF.DATA data sets and you can enter these statements in mixed case.

NPFPRINTPREFIX specifies a high-level qualifier to be used for NPF's temporary QSAM print data sets.

NPFJESALLOCATION defines the type of space being allocated and primary and secondary sizes of the data sets in an NPF JES implementation. It corresponds to the JCL SPACE parameter.

NPFVTAMALLOCATION defines the type of space being allocated and primary and secondary sizes of the data sets in an NPF VTAM implementation. It corresponds to the JCL SPACE parameter.

NPFQMGRTHREAD specifies the number of LPRs running under the NPF queue manager application. The default is 4.

NPFUNIT specifies the UNIT name for the dynamic allocation of the QSAM data sets. It corresponds to the JCL UNIT parameter.

NPF.DATA sample

The following NPF.DATA sample is provided in *hlq*.SEZAINST(NPFDATA). Copy this sample to your library and modify it to suit your installation.

```
;
;  COPYRIGHT = NONE.
;
NPFPRINTPREFIX    NPF.TEMP
NPFJESALLOCATION  40000,2,2
NPFVTAMALLOCATION TRK,1,1
NPFQMGRTHREAD     5
NPFUNIT           SYSDA
```

Figure 3. NPF.DATA sample

NPF.DATA search order

To find NPF.DATA, NPF follows a standard search sequence. It reads each instance of NPF.DATA it can find, in order, until it finds values for all the NPF configuration statements. It uses the first value that it finds for each statement. If an allocation fails, the data set does not exist, or the data set is not available, NPF goes to the next data set in the sequence. The search ends when it finds values for all NPF configuration statements or when all data sets have been checked.

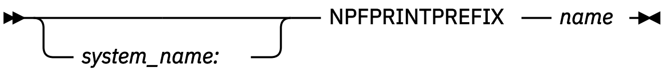
The search order is:

1. The data set pointed to by a DD statement //SYSNPFD.
2. A data set with the name of *jobname*.NPF.DATA, where *jobname* is the jobname for the NPF capture point application or the user ID for the TSO session (panels).
3. A data set with the name of SYS1.TCPPARMS(NPFDATA).
4. A data set with the name of *hlq*.NPF.DATA, where *hlq* is the value specified by the DATASETPREFIX statement in TCPIP.DATA. If TCPIP.DATA cannot be found or does not contain a DATASETPREFIX statement, then *hlq* is the system default, TCPIP.

NPFPRINTPREFIX statement

The NPFPRINTPREFIX statement specifies a high-level qualifier of 26 characters or fewer used for NPF print data sets.

Syntax



Parameters

***system_name*:**
If you are sharing the NPF.DATA file across more than one MVS system, this parameter specifies the MVS system name. This value should be set to the same name as your JES NJE *nodename*. The colon is required.

name
A high-level qualifier for NPF print data sets. The name can be up to 26 characters and must follow these rules:

- It must not start or end with a period.
- Each qualifier segment must be no more than 8 characters long and consist of a combination of alphabetic, numeric, and some special characters. (See the *MVS JCL Reference* for details.)
- It must start with an alphabetic character, and each period must be followed by an alphabetic character.
- All alphabetic characters are converted to uppercase.

Usage notes

- If this statement is not specified, the system uses the general TCP/IP high-level qualifier, either specified via the DATASETPREFIX statement, or defaulted to TCPIP.
- Any high-level qualifier longer than 15 characters might cause truncations of the major and minor names used as part of the data set name. This applies to the NPFPRINTPREFIX value or, if one is not specified, it applies to the TCP/IP default high-level qualifier.

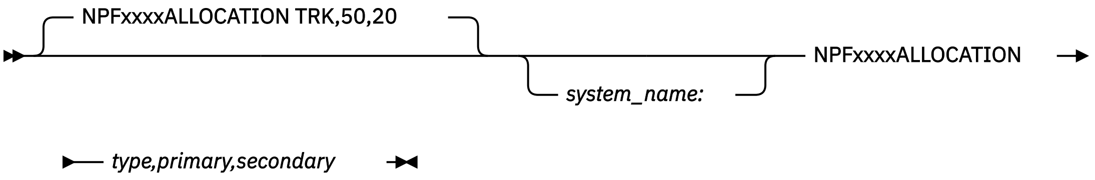
Related topics

See the *OS/390® Program Directory* for information about running EZAPPRFX.

NPFxxxxALLOCATION statement

The NPFxxxxALLOCATION statement specifies space size for either JES or VTAM that corresponds to the JCL SPACE parameter. Use JES or VTAM to replace xxxx in the statement. Specify the type of space being allocated and the primary and secondary sizes of the files.

Syntax



Parameters

***system_name*:**
If you are sharing the NPF.DATA file across more than one MVS system, this parameter specifies the MVS system name. This value should be set to the same name as your JES NJE *nodename*. The colon is required.

type
The type of allocation: CYL, TRK, or a number specifying a block size if the allocation is to be done in blocks. The default is TRK.

primary
The size of the primary allocation for the files. If CYL is specified as the type, this specifies the number of cylinders. If TRK is specified as the type, this specifies the number of tracks. If a block size is specified as the type, this specifies the number of blocks. The default is 50.

secondary
The size of the secondary allocation for the files. If CYL is specified as the type, this indicates the number of cylinders. If TRK is specified as the type, this indicates the number of tracks. If a block size is specified as the type, this indicates the number of blocks. The default is 20.

Examples

Examples of this specification are:

```
NPFJESALLOCATION CYL,1,1
```

```
NPFVTAMALLOCATION TRK,1,1
```

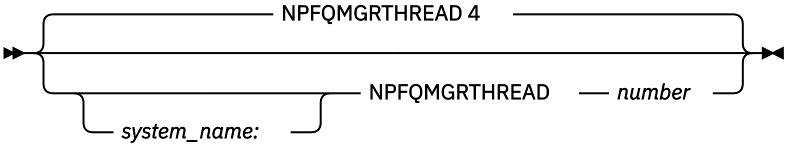
Usage notes

- The values for the type, primary, and secondary allocations must be separated by commas.
- Specify primary extents large enough for the majority of jobs. Unused primary space is released after the data set is created.
- Specify non-zero secondary extents to ensure enough space to hold your largest data set.

NPFQMGRTHREAD statement

The NPFQMGRTHREAD statement specifies the number of LPRs running under the NPF queue manager application.

Syntax



Parameters

***system_name*:**
If you are sharing the NPF.DATA file across more than one MVS system, this parameter specifies the MVS system name. This value should be set to the same name as your JES NJE *nodename*. The colon is required.

number
A positive number, from 1 to 11, specifying the number of LPRs running under the NPF queue manager application. The default is 4.

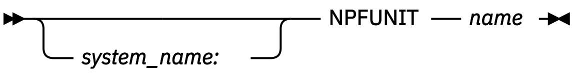
Usage notes

The number of LPR threads is limited by the amount of below-the-line virtual storage available. Each LPR thread requires approximately 0.75 MB of below-the-line storage. If virtual storage problems exist, reduce the number of threads specified on this statement.

NPFUNIT statement

The NPFUNIT statement specifies a unit name of 8 or fewer characters to be used for the dynamic allocation of the QSAM print data sets.

Syntax



Parameters

***system_name*:**
If you are sharing the NPF.DATA file across more than one MVS system, this parameter specifies the MVS system name. This value should be set to the same name as your JES NJE *nodename*. The colon is required.

name
A UNIT name for the DASD volume(s) where NPF data sets will be allocated. The name must follow these rules:

- It must define an esoteric unit name (for example, SYSDA), a generic unit name (for example, 3380), or a specific unit address.
- It must be at least one character and no more than 8 characters long. It consists of a combination of alphabetic, numeric, and some special characters.
- It must not start or end with a period.

Examples

Examples of this specification are:

```
NPFUNIT SYSDA
```

```
NPFUNIT 3380
```

Usage notes

If this statement is not specified, NPF uses the default value of SYSDA.

Customization for JES output processing

The Network Print Facility uses the functional subsystem interface (FSI) to communicate with MVS JES. The NPF functional subsystem (FSS) writer processes each job on the JES output queue by JES output class, forms, and destination information. The DEST, CLASS, and FORMS parameters are used to create the record key for the routing file in JES implementations and, therefore, play an important role in determining where to print JES output.

The basic concepts for JES output processing are explained in "Network Print Facility interfaces with JES" on page 5. Chapter 8, "Operating the Network Print Facility," on page 111 provides additional information for defining the NPF FSS writer and tailoring it to your particular installation.

This section describes how to:

- Customize JESNEWS
- Define and authorize an NPF FSS writer
- Determine where to print JES output

Be sure to see "Checklist of the NPF tasks for JES users" on page 133 to verify that you have covered all the JES planning and customization tasks.

Customizing JESNEWS

JESNEWS is a separate data set that might be sent across the FSI to FSS printers under certain circumstances. It contains installation dependent information. Installations define the JESNEWS text via a DSP in JES3 (called JESNEWS) or via a particular external writer name in JES2 (called 'JESNEWS', that you write to using IEBGENER or IEBDG.)

For JES3, JESNEWS appears if an FSS writer is producing BURST pages. Ensure you specify BURST=NO, so the production of JESNEWS data sets does not affect NPF processing.

For JES2, JESNEWS appears if you specify SEP=YES and set up JESNEWS. JES2 exit 23 can be used to turn JESNEWS off so as not to affect NPF processing.

JES3 users can see *JES3 Commands* for details about how to use the JESNEWS DSP. JES2 users can find out more about JESNEWS in the *JES2 Initialization and Tuning Guide.*

Defining the NPF FSS writer

FSS writers control the processing of JES logical printers.The number of NPF FSS writers and the number of logical printers associated with each NPF FSS writer is installation-dependent. Each FSA subtask under the FSS writer acts as a logical printer dedicated to processing the output in its defined class or classes. The steps to define FSS writers and their logical printers vary depending on the version and release of JES.

Because each FSA subtask acts as a logical printer, after the FSA subtask has been defined, each instance of a writer can be started and stopped like any printer. The FSS writer can be started by JES commands or through the JES initialization definition. Commands and initialization statements to start the FSS writer in various environments can be found in Table 13 on page 115.

Authorizing the NPF FSS writer load module

The FSS load module (EZAPPFS) must be marked as both authorized and reusable, and placed in an APF-authorized library. To ensure subpools 0-127 are in the correct key, place EZAPPFS in the MVS program properties table (PPT) with the following entry:

```
PPT  PGMNAME(EZAPPFS) KEY(1)
```

Note: PPT entries in the SCHED*xx* member of SYS1.PARMLIB are no longer required. They are supplied in the default PPT table via load module IEFSDPPT.

Determining where to print JES output

Each NPF FSS writer is associated with 1 or more logical printers (which was assigned an output class or classes during JES initialization, or in some cases, by JES console commands). When an NPF FSS writer and associated logical printer are active, the NPF FSS writer processes all JES output that matches the output classes assigned to the logical printers. When the Network Print Facility receives the output, it uses the output class, destination, and forms associated with the output to determine LPR options and which LPD printer queue should receive the output. This determination is made though the routing file described in the following chapter.

For example:

```
Output 1 has JCL parameter CLASS=C, FORMS=1000, DEST=LOCAL
Output 2 has JCL parameter CLASS=C, FORMS=1000, DEST=LOCAL
Output 3 has JCL parameter CLASS=C, FORMS=1000, DEST=DEST1
Output 4 has JCL parameter CLASS=C, FORMS=2000, DEST=LOCAL
```

Output 1 and 2 will be sent to the same LPD print queue, which were defined to handle the combination of the class, forms, and destination. Output 1, 3, and 4 might each be sent to separate printers or might have a different NPF processing performed because they each have a unique combination of class, forms, and destination. For more information on printer destination mapping, see "The routing file" on page 38.

Note: The Network Print Facility uses the 1–8 alphanumeric destination name you specify through the ROUTING function of the EZAPPFL macro (MAJKEY parameter) to route output.

Defaults for JES output data

The Network Print Facility uses the following defaults for output data sets:

Table 3. JES data limitations

Data Type	NPF Default
line mode data	LRECL = 4092, BLKSIZE = 4096
stream mode data	LRECL = 32 756, BLKSIZE = 32 760

Note: The maximum record length the Network Print Facility processes for line mode data is 4092.

All JES output data sets should be coded as follows:

RECFM=VBA for ASA carriage control
RECFM=VBM for machine carriage control
RECFM=VB for no carriage control

Customization for VTAM output processing

The Network Print Facility uses an NPF VTAM capture point application to process VTAM output. This section describes activities to ensure correct processing. The topics included are:

- Consistent resource definition
- File definition
- The EZAPPEFM macro
- Page format definition
- The EZAPPDPF macro

Be sure to see "Checklist of the NPF tasks for VTAM users" on page 132 to verify that you have covered all the VTAM planning and customization tasks.

See "Starting the NPF VTAM capture point" on page 117 for the procedure to start the NPF VTAM capture point application.

Creating a startup procedure

You can specify the amount of storage through the procedure to start the VTAM capture point application.

Procedure

Follow these steps to estimate your storage needs and then specify them in the startup procedure.

1. Determine the amount of below-the-line private storage needed for the NPF VTAM capture point application. Consider the following factors when estimating this storage requirement:

   a) Storage for a minimum-configuration application:

For an NPF VTAM capture point application, the simplest useful configuration would have one logical printer LU running one session with all output routed to a single destination. The amount of below-the-line storage needed for such a configuration varies from one MVS system to the next.

- For preliminary planning purposes, assume 0.5 MB will be needed for this configuration.
- **Tuning note**: After NPF is installed, a simple test run with this configuration will enable you to refine that estimate. The JCL output listing from that test will show the below-the-line private storage actually used on your system.

b) Storage for active print files:

Print files being processed by the NPF VTAM capture point application are managed as temporary QSAM data sets requiring below-the-line storage.

- Each file being received requires 16 KB.

Specify either the MAXFLSTG or MAXOPEN JCL parameter to limit the amount of below-the-line storage used for open print data sets. With either of these parameters, if a request for a new data set would cause that limit to be violated, that request will be rejected with a sense code indicating you had insufficient resources. For example, a limit of 1000 open files could be expressed by either MAXFLSTG=16000K or MAXOPEN=1000.

c) Other minor storage requirements:

- Each logical printer LU requires 112 bytes of below-the-line control block storage.
- Each session requires 112 bytes of below-the-line control block storage.

2. Set up the JCL to run the VTAM capture point application using the estimates calculated in the previous step. Include the following parameters on the EXEC statement.

a) Identify the program to be started.

```
PGM=EZAPPAAA
```

b) Code the REGION parameter to specify the amount of below-the-line storage calculated in the previous step. The region sizedepends on the number of active sessions and the maximum number of open print files in progress at any given time. This should include your minimum configuration requirement plus the maximum storage for active print files and other minor storage requirements.

c) Code the PARM parameter with LUCLASS, and either MAXFLSTG or MAXOPEN.

```
  ,PARM=(LUCLASS=(luclass_list),
         MAXFLSTG=file_stg_limit,
         MAXOPEN=open_file_limit)
```

Specify *luclass_list* as one or more LU class numbers with values from 1 to 64. Specify *file_stg_limit* as the maximum amount of buffer space you want to allow for open print data sets, or specify *open_file_limit* as the maximum number of open print data sets.

These LU class numbers will be used as follows to select the set of logical printer names that the application represents:

- For each VTAM-usable entry in the routing file, the MAJKEY keyword must specify a logical printer name, and the LUCLASS value must specify one or more LU class numbers with values from 1 to 64.
- The LU class data from the JCL will be compared to the LU class data in each routing file entry. If there is a match on at least one LU class number, the application will represent the logical printer name specified by that entry.

For example, assume that the routing file contains 4 entries as follows:

PRINTERA LUCLASS=(1,4)
PRINTERB LUCLASS=(2,3,4)
PRINTERC LUCLASS=(3,4)
PRINTERD LUCLASS=(2,4)

Given the above routing file entries, code the JCL LUCLASS parameter as follows:

To use all 4 printers
LUCLASS=(4)

To use printers B, C, and D,
LUCLASS=(2,3)

To use printers B and D
LUCLASS=(2)

To use just printer A
LUCLASS=(1)

See "Starting the NPF VTAM capture point" on page 117 for the sample JCL and a further explanation of the LUCLASS, MAXOPEN, and MAXFLSTG parameters.

d) To prevent MVS from terminating the application because of its running time, code the JOB statement with `TIME=1440` or `TIME=NOLIMIT`.

Defining consistent resources

Consistent definition of resources is essential to use the NPF VTAM capture point application.

Procedure

To ensure consistent definition for VTAM customization, perform these tasks ***before*** starting the Network Print Facility. See z/OS Communications Server: SNA Resource Definition Reference for additional detail on these items.

1. Set up an APPL definition for each logical printer name.

   Define each of the logical printer names to VTAM by an APPL definition statement in an application major node definition. Selected parameters on that definition statement are discussed below:

   a. If any partner LU will need accurate BIND parameter information from the logon mode table, add a new table entry or find an existing one that meets the requirements. Then specify the necessary table name and entry name through the MODETAB and DLOGMOD keywords, or set up the table so the entry that you want will be selected by default.

   b. Code SESSLIM=YES to force a session limit of 1 for the logical printer.

   c. Specify (or default to) PARSESS=NO.

   d. Do not code any APPC-related keywords.
2. In each session partner LU, set up definitions for each logical printer name. For CICS, a specific requirement is that you specify NO for the QUERY keyword on the TYPETERM macro. These logical printer definitions should correspond to the VTAM APPL names.
3. Build the routing file to make the connection between the logical printer names and the LUCLASS numbers. Specify the logical printer LU name in the first part of the routing record key (Major Name) and specify the printer classes in the LUCLASS field. (See the "Routing file input fields" on page 39.)

Modifying the program properties table

To run the NPF VTAM capture point application as nonswappable, add a Network Print Facility entry for the VTAM application, EZAPPAAA, in the MVS program properties table (PPT).

```
PPT  PGMNAME(EZAPPAAA) NOSWAP
```

Note: PPT entries in the SCHED*xx* member of SYS1.PARMLIB are no longer required. They are supplied in the default PPT table via load module IEFSDPPT.

File definition for VTAM output

The print data for a given logical printer LU is received as a stream of request units (RUs). The SNA architecture defines two groupings of RUs that are of interest here:

- A ***chain*** consists of one or more RUs.
- A ***bracket*** consists of all of the RUs in one or more related chains.

For each logical printer, the Network Print Facility divides the stream of received RUs into a series of ***print files*** and queues the data for each print file in its own sequential data set, keeping it together as it is transmitted to LPD and printed.

For printing within the SNA network, there is no need for this print file concept. An application starting a session with a printer gains exclusive control of that printer for the duration of the session, and data is printed as soon as it arrives at the printer. Therefore, data is printed in the correct order, and there is no danger of intrusion by any other application's print output.

In contrast, when printing in the TCP/IP network, the sending application is not guaranteed exclusive control of the printer. All data within a single file is printed together in the correct order. However, between two successive files from one application, there is nothing to prevent the printing of files from other applications. Therefore, to guarantee that data belonging together actually gets printed without intrusions from other sources, it is important to batch that data together into a single file.

From an SNA-theory point of view, the use of brackets is the most logical way to define file boundaries. Using this technique, a request marked BB (Begin Bracket) indicates the start of a new file, and the end of a chain marked EB (End Bracket) marks the end of the file. This is the default file definition technique used by the Network Print Facility. It works with both CICS and IMS LU type 1 sessions, with CICS LU type 3 sessions, and with other applications able to control their use of the BB and EB bracket bits.

There are situations where using bracket bits does not work well, as indicated by the following examples:

1. CICS application with an LU type 0 session:
   - BB is sent on the session's first request.
   - No EB is sent at the end of the transaction.
   - Result: The output for the entire session is treated as one file, causing a potentially long delay in printing.
2. IMS application with a non-SNA 3270 printer:
   - Each line of output is sent as a separate chain marked BB, EB.
   - Result: Each line is treated as a separate file, causing unnecessary file overhead and increasing the chances of other files from other applications intruding into the sequence.

End-of-File definition

The NPF VTAM capture point application provides 5 different end-of-file rules for defining files, as explained below. Only one of these rules can be specified for any given session:

1. End-of-file = end-of-bracket (the default rule).
2. End-of-file = end-of-session.
3. End-of-file = end-of-chain.
4. End-of-file indicated by a specified string of data in the file's last end-of-chain request.
5. End-of-file indicated by the expiration of a timer.

   Timers should be used only as a last resort in situations where none of the other end-of-file rules make sense.

   - The use of timers to detect file boundaries is an inexact science. Experimentation and tuning is required to choose the best timer values, and even then the results will not be perfect.

- Under each of the other end-of-file rules, exact file boundaries can be determined directly from the received print data requests.

The Network Print Facility uses a table to allow user specification of the end-of-file rules for different sessions. This user-replaceable table is defined using the EZAPPEFM macro and contains one or more entries. Each entry defines the end-of-file rules used with various combinations of PLU name and LU type. The table entry is selectable on a per-SLU basis, via the end-of-file entry name in the SLU's routing file entry. This end-of-file rules table is packaged in the module EZAPPEFT. That module is shipped in two different forms:

- As a ready-to-use load module which, if used as is, causes the end-of-file = end-of-bracket rule to be used on all sessions.
- As sample source code, which can be modified by the user to specify different rules in selected cases, then assembled and used instead of the IBM-supplied module.

Installing the end-of-file table

This end-of-file rules table is packaged in module EZAPPEFT. The EZAPPEFT is shipped in the form as a ready-to-use module or as sample source code. The users can modify the sample source code form.

Procedure

To install a user-modified version of EZAPPEFT, proceed as follows:

1. Code the modified version of the module, using the IBM-supplied EZAPPEFM macro. See "Defining end-of-file rules" on page 27 for details about how to use the macro to create the table.
2. Assemble and link-edit the module into a partitioned data set different from that used for the IBM-supplied version of the module. This prevents the module from being deleted during any subsequent re-install of NPF.
3. On the JCL for the NPF VTAM capture point application, provide a STEPLIB DD statement specifying that partitioned data set. For example:

   ```
   //STEPLIB DD DSN=xtable.loadlib, DISP=SHR
   ```

   In this example, `xtable.loadlib` is the name of the partitioned data set that includes the tables as members. See "Operating the NPF VTAM capture point" on page 117 for a sample startup procedure.

Results

After performing these steps, the new table will automatically replace the IBM-supplied version the next time the NPF VTAM capture point application is started.

To dynamically replace the current version of EZAPPEFT with a new one while the NPF VTAM capture point application is running, respond to message EZY0837I with the following code:

```
nn,RELOAD EZAPPEFT
```

After the dynamic replacement is complete, the new version of the table will be used for new sessions, but already-started sessions will continue to run under the rules from the old table.

See "RELOAD: Dynamically replacing a table" on page 120 for more information about RELOAD.

References to the end-of-file table from the routing file

Each SLU's routing file record contains the name of the EZAPPEFT table entry to be used for sessions involving that SLU. These table entry name references in the routing file can be explicitly coded through an ISPF panel or through an assembly using the EZAPPFL macro. If you do not code a table entry reference, the Network Print Facility sets up the routing file record to see the table entry named DFLTNTRY.

The structure of the end-of-file rules table (EZAPPEFT)

The statements to generate module EZAPPEFT should contain nothing except EZAPPEFM macros and comments. No other Assembler language statements should be coded. It should include at least one EZAPPEFM macro with TYPE=ENTRY and must end with exactly one EZAPPEFM macro with TYPE=END. Additional EZAPPEFM macros can be coded as needed.

IBM recommends that one EZAPPEFM macro with TYPE=ENTRY be coded with the macro label DFLTNTRY. This is the table entry name that is generated by default in any routing file record when the user does not specify an EOFNAME value in the ROUTING function of the EZAPPFL macro or an EOFILE NAME in an ISPF panel (see Figure 27 on page 79).

Defining end-of-file rules

You can code the EZAPPEFM macro to define end-of-file rules for each combination of SLU name, PLU name, and LU type. There are 4 types of EZAPPEFM macros you can code. TYPE=*value* is a required keyword identifying the function of the macro. Choose one of 4 valid values, as follows:

TYPE=ENTRY
Use this macro type to start a new table entry and optionally define entry-wide end-of-file defaults for one or more LU types. This function of the EZAPPEFM macro requires that you specify the name for the end-of-file rules table entry, *maclabel*. Do not code *maclabel* with any of the other types of EZAPPEFM (TYPE=SYSDFLT, TYPE=PLU, or TYPE=END).

TYPE=PLU
Use this macro to define end-of-file rules for one or more LU types when used with a specified PLU or set of PLUs. This macro type is optional. If used, several can follow an EZAPPEFM macro with TYPE=ENTRY.

TYPE=SYSDFLT
Use this macro type to define system-wide end-of-file defaults for one or more LU types. This macro type is optional. If used, there can be only one, and it must be the first EZAPPEFM macro in the module.

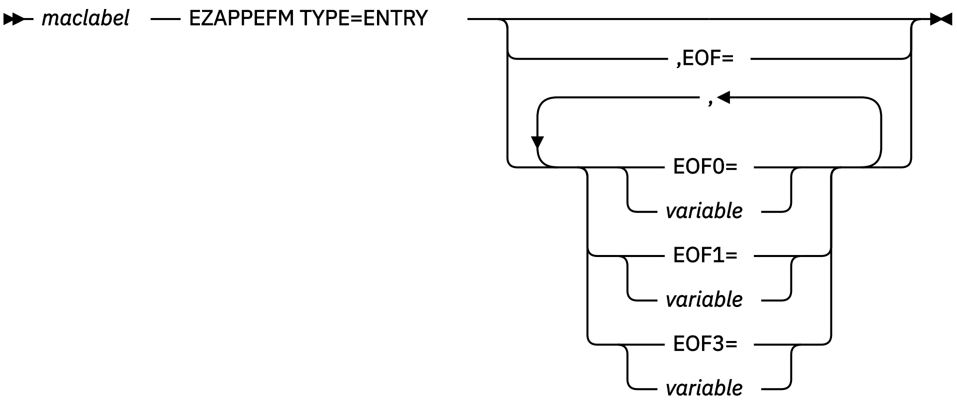
TYPE=END
Use this macro type to mark the end of the table. You must code exactly one EZAPPEFM macro with TYPE=END, and it must be the last EZAPPEFM macro in the module.

Each macro type is described separately, with separate syntax diagrams.

EZAPPEFM TYPE=ENTRY

TYPE=ENTRY indicates that this macro starts a new table entry and optionally defines entry-wide end-of-file defaults for one or more LU types.

Syntax



Parameters

maclabel

When you code TYPE=ENTRY, you must use *maclabel* to specify the name for the end-of-files rules table entry. Each table entry name must be unique and must conform to the rules for labels in assembler language. This name correlates to what the user specifies as the EOFNAME keyword in the ROUTING function of the EZAPPFL macro or as the EOFILE NAME field on the ISPF panel. See Figure 27 on page 79.

EOF=

Specifies an end-of-file rule for all LU types.

EOF0=, EOF1=, and EOF3=

Specifies an end-of-file rule for the single LU type 0, LU type 1, or LU type 3. The EOF0=, EOF1=, and EOF3= keywords can be coded together but cannot be used if the EOF= keyword is coded.

variable

The following variables can be coded for EOF, EOF0, EOF1, or EOF3:

EB

Indicates end-of-bracket.

EC

Indicates end-of-chain.

ES

Indicates end-of-session.

(STRING,KEEP,*string*)

Indicates a character or hexadecimal end-of-file data string that remains as part of the print data. The value of *string* is coded as either C'character string' or X'hexadecimal string' and cannot exceed 56 bytes in length. The entire string must be received within a single RU, and that RU must be marked last-in-chain.

(STRING,DEL,*string*)

Indicates a character or hexadecimal end-of-file data string that is deleted prior to printing. The value of *string* is coded as either C'character string' or X'hexadecimal string' and cannot exceed 56 bytes in length. The entire string must be received within a single RU, and that RU must be marked last-in-chain.

(TIMER,*idleint,busyint*)

Indicates that files are closed according to the *idleint* and *busyint* values. This choice is allowed ***only*** with the EOF0 and EOF3 keywords for LU type 0 and LU type 3.

idleint

Required with TIMER. Indicates the idle time interval, in seconds. An open file is closed if this time passes without any more input being received and the current file state indicates a reasonable stopping point (between chains).

busyint

Required with TIMER. Indicates the busy time interval, in seconds. An open file is closed at the next reasonable stopping point (between chains and at the top of a new page) if this time passes without any expirations of the idle interval timer. This interval prevents delays in printing output when a given session is sending print output almost continuously.

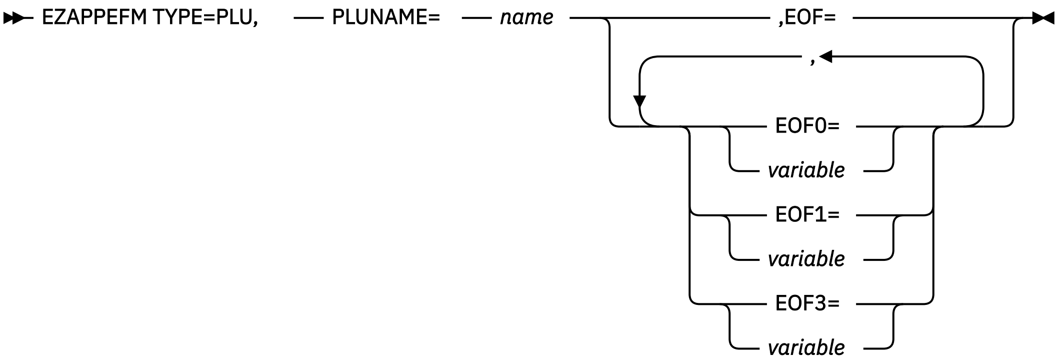
Usage notes

- With TYPE=ENTRY, use of the EOF or EOFx keywords is optional.
- For an SLU using this table entry, each of these keywords defines a default rule for all sessions using the specified LU types.

EZAPPEFM TYPE=PLU

TYPE=PLU indicates that this macro defines end-of-file rules for one or more LU types when used with a specified PLU or set of PLUs. Use of an EZAPPEFM macro with TYPE=PLU is optional. Several can be coded following an EZAPPEFM macro with TYPE=ENTRY.

Syntax



Parameters

name

Identifies a set of PLU names for which special end-of-file rules should be used instead of the default rules specified (or set by default) for the table entry. The following variations are allowed for the 1 to 8 byte name:

- The full name of a specific PLU.
- The partial name of a family of PLUs, with unspecified information replaced with * or ? characters.

EOF=, EOF0=, EOF1=, and EOF3=

At least one of these keywords is required. For an SLU using this table entry, each of these keywords defines a rule to be used for the specified LU types during any session in which the PLU name matches the macro's PLU name specification.

See the description of TYPE=ENTRY for the valid values of the EOF, EOF0, EOF1, and EOF3 *variable* and their coding rules.

Usage notes

Follow these rules when coding the PLUNAME:

- A single * represents any number of prefix or suffix characters. Using * in the middle of a partial PLU name is not allowed.
- A single ? represents exactly one character anywhere within the name.
- Do not mix * and ? characters in the same PLU name.
- There must be at least 1 significant character (other than * or ?) in the PLU name.

Examples

- Specify any PLU name starting with ABC:

```
PLUNAME=ABC*
```

- Specify any PLU name ending with ABC:

```
PLUNAME=*ABC
```

- Specify any PLU name containing the string ABC anywhere within the name:

```
PLUNAME=*ABC*
```

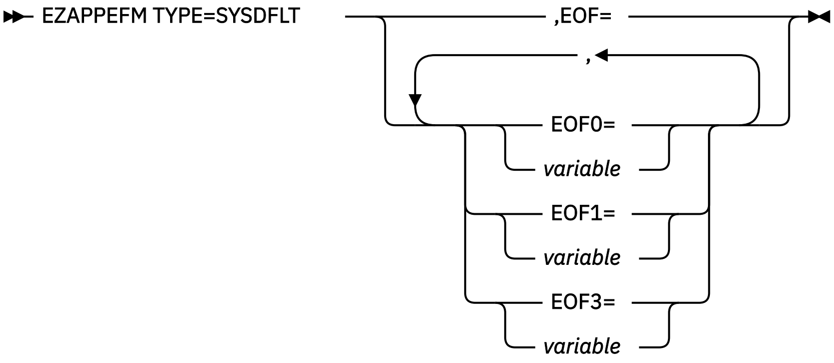
- Specify any PLU name containing the string AB in bytes 2 and 3 and the character C in byte 7:

```
PLUNAME=?AB???C?
```

EZAPPEFM TYPE=SYSDFLT

TYPE=SYSDFLT indicates that this macro defines system-wide end-of-file defaults for one or more LU types. Use of an EZAPPEFM macro with TYPE=SYSDFLT is optional; if coded, there can be only one, and it must be the first EZAPPEFM macro in the module.

Syntax



Parameters

EOF= EOF0= EOF1= EOF3=

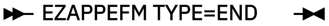
At least one of these keywords is required. Each of these keywords defines a rule to be used as the system-wide default for the specified LU types.

See the description of TYPE=ENTRY for the valid values of the EOF, EOF0, EOF1, and EOF3 *variable* and their coding rules.

EZAPPEFM TYPE=END

TYPE=END indicates that this macro marks the end of the table. You must code exactly one EZAPPEFM macro with TYPE=END, and it must be the last EZAPPEFM macro in the module. With TYPE=END, no keywords are allowed.

Syntax



Parameters

None

End-of-file table example

Figure 4 on page 31 shows an example of a user-modified end-of-file table. It is more extensive than what might be needed at an actual installation and does not resemble the IBM-supplied source code. The notes in the table are explained in “Example notes” on page 31.

```
***********   Override of system default   ***********
* See note 2.
*
         EZAPPEFM TYPE=SYSDFLT,                                       X
               EOF0=(TIMER,15,900)
*
************   Start of 1st table entry   ************
* See note 3.
*
DFLTNTRY EZAPPEFM TYPE=ENTRY
*
* See note 5a.
*
         EZAPPEFM TYPE=PLU,                                           X
               PLUNAME=SPECAPL1,                                      X
               EOF=ES
*
* See note 5b.
*
         EZAPPEFM TYPE=PLU,                                           X
               PLUNAME=IMSA*,                                         X
               EOF3=(TIMER,10,10800)
*
* See note 5c.
*
         EZAPPEFM TYPE=PLU,                                           X
               PLUNAME=IMS*,                                          X
               EOF3=(TIMER,15,900)
*
*
************   Start of 2nd table entry   ************
* See note 4.
*
ENTRY2   EZAPPEFM TYPE=ENTRY,                                         X
               EOF3=(STRING,KEEP,C'End of Print Job')
*
* See note 6a.
*
         EZAPPEFM TYPE=PLU,                                           X
               PLUNAME=SPECAPL2,                                      X
               EOF1=EC
*
* See note 6b.
*
         EZAPPEFM TYPE=PLU,                                           X
               PLUNAME=??XYZ???,                                      X
               EOF0=(STRING,DEL,C'<<Temporary EOF String>>'),         X
               EOF3=(STRING,DEL,C'<<Temporary EOF String>>')
*
************       End of the table       ************
*
         EZAPPEFM TYPE=END
```

Figure 4. Example of an End-of-File table

Example notes

1. Assembler source statements CSECT, RMODE, AMODE, and END are generated by the EZAPPEFM macros and, therefore, should not be coded by the user.
2. The EZAPPEFM macro with TYPE=SYSDFLT changes the system-wide default rule for LU0 sessions only. Under the new default rule for LU0, end-of-file is determined by the timer, with an idle interval of 15 seconds and a busy interval of 900 seconds (15 minutes).

   For LU1 and LU3 sessions, nothing is specified; therefore, the normal system-wide default rules, EOF1=EB and EOF3=EB, are used for those LU types.
3. The EZAPPEFM macro with TYPE=ENTRY and the label DFLTNTRY defines the start of the first table entry. This entry is used by any SLU whose routing file record set up specified no end-of-file entry name or the entry name DFLTNTRY.

   This macro does not contain any EOF*x*= keywords. Therefore, the default rules applying to this table entry are the same as the system-wide default rules explained above.

4. The EZAPPEFM macro with TYPE=ENTRY and the label ENTRY2 defines the start of the second table entry. This entry is used by any SLU whose routing file record specifies ENTRY2 as its end-of-file entry name.

   This macro contains an EOF3= keyword, which overrides the system-wide default rule for LU3 sessions. Under the default rule, end-of-file is marked by the character string `End of Print Job`, and that string should be left as part of the print data.

   For LU0 and LU1 sessions, nothing is specified; therefore, the system-wide default rules, EOF0=(TIMER,15,900) and EOF1=EB, apply to this table entry.
5. For all SLUs using the entry named DFLTNTRY, the following end-of-file rules are used:
   a. If the PLU name is SPECAPL1, the print output for the entire session is treated as a single file for all LU types.
   b. If the PLU name starts with IMSA:
      - On LU3 sessions, end-of-file is determined by the timer, with an idle interval of 10 seconds and a busy interval of 3 hours.
      - On LU0 sessions, the system-wide default rule, EOF0=(TIMER,15,900), is used
      - On LU1 sessions, the system-wide default rule, EOF1=EB, is used
   c. If the PLU name does not start with IMSA but starts with IMS:
      - On LU3 sessions, end-of-file is timer-determined, with an idle interval of 15 seconds and a busy interval of 15 minutes.
      - On LU0 sessions, the system-wide default rule, EOF0=(TIMER,15,900), is used
      - On LU1 sessions, the system-wide default rule, EOF1=EB, is used
   d. For all other PLU names, the Network Print Facility uses the default rules for the table entry.
6. For all SLUs using the entry named ENTRY2, the following end-of-file rules are used:
   a. If the PLU name is SPECAPL2:
      - On LU1 sessions, each chain is treated as a separate file.
      - On LU0 sessions, the system-wide default rule, `EOF0=(TIMER,15,900)` is used
      - On LU3 sessions, this table entry's default rule, `EOF3=(STRING,KEEP,C'End of Print Job')`, is used
   b. If the PLU name contains XYZ for its 3rd thru 5th bytes:
      - On LU0 and LU3 sessions, end-of-file is marked by the string `<<Temporary EOF String>>`, and that string is deleted before the file is printed.
      - On LU1 sessions, the system-wide default rule, EOF1=EB, is used.
   c. For all other PLU names, the Network Print Facility uses the default rules for the table entry.

Page format definition

In order to correctly handle the SCS print data received over an LU type 1 session, NPF must use a page format definition that is consistent with the target LPD printer. At session start-up, NPF determines the default page format for the session. As print data is processed, SCS commands within the print data stream might override some or all of the default page format information.

NPF uses a table to allow user specification of default page formats for different sessions. This user-replaceable table is defined using EZAPPDPF macros and contains one or more entries with both horizontal and vertical page format data. A table entry is selectable on a per-SLU basis, via the default-page-format entry name in the SLU's routing file record.

The default page format table is packaged in the module EZAPPPFT. The EZAPPPFT module is shipped in two different forms:

- As a ready-to-use load module which contains an empty table. If used as is, this table causes NPF's hard-coded default page format values to be used for all sessions.
- As sample source code, which can be modified by the user to specify different page formats in selected cases, then assembled and used instead of the IBM-supplied module.

Installing the page format table

The default page format table is packaged in the module EZAPPPFT. The EZAPPPFT module is shipped in the form as a ready-to-use load module or as sample source code. The sample source code form can be modified by the user.

Procedure

To install a user-modified version of EZAPPPFT, proceed as follows:

1. Code the modified version of the module, using the IBM-supplied EZAPPDPF macro. See "EZAPPDPF" on page 34 for details about how to use the macro to create the table.
2. Assemble and link-edit the module into a partitioned data set different from that used for the IBM-supplied version of the module. This prevents the module from being deleted during any subsequent re-install of NPF.
3. On the JCL to start the VTAM capture point application, provide a STEPLIB DD statement specifying that partitioned data set. For example:

   ```
   //STEPLIB DD DSN=xtable.loadlib, DISP=SHR
   ```

   In this example, `xtable.loadlib` is the name of the partitioned data set that includes the tables as members. See "Operating the NPF VTAM capture point" on page 117 for a sample startup procedure.

Results

To dynamically replace the current version of EZAPPPFT with a new one while the NPF VTAM capture point application is running, respond to message EZY0837I with the following code:

```
nn, RELOAD EZAPPPFT
```

After the dynamic replacement is complete, the new version of the table will be used for new sessions, but already-started sessions will continue to run under the rules from the old table.

See "RELOAD: Dynamically replacing a table" on page 120 for more information about dynamically replacing a table.

References to the page format table from the routing file

Each SLU's routing file record optionally contains the name of the EZAPPPFT table entry to be used for sessions involving that SLU. These table entry name references in the routing file can be explicitly coded through the DEF PAGE FORMAT field in an ISPF panel (see Figure 27 on page 79) or through an assembly using the DFPNAME field of the EZAPPFL macro. If you do not code a table entry reference, the Network Print Facility sets up the routing file record with its table entry name set to blanks.

Determining the default page format

During session initiation, NPF checks the SLU's routing file record to see if a page format table entry name has been specified. Processing occurs as follows:

- If an entry name has been specified, the default page format data is copied from the named table entry. (If the named entry is not found, the session is rejected.)

- If the entry name is blank but the table contains an entry named DFLTNTRY, the default page format data is copied from that entry.
- If the entry name is blank and the table does not have an entry named DFLTNTRY, NPF uses hard-coded default page format values of MPP=80, MPL=1, LM=1, RM=80, TM=1, BM=1. (See "EZAPPDPF" on page 34 for more information.)

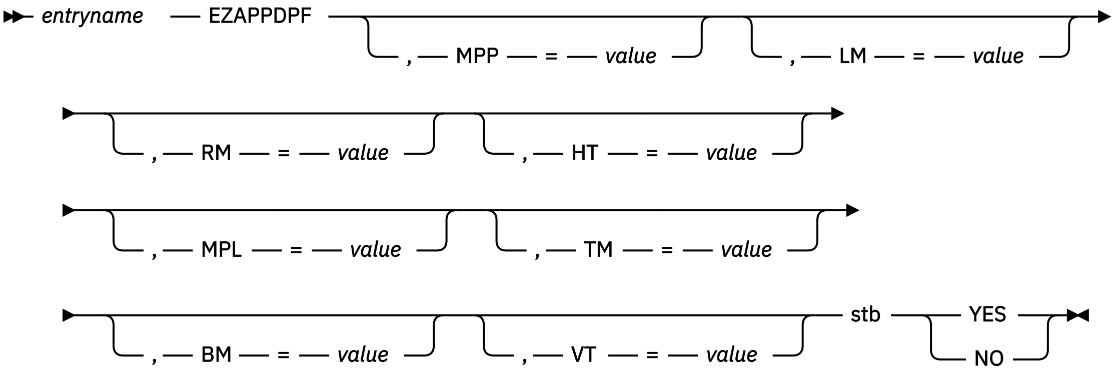
The structure of the page format table (EZAPPPFT)

The page format table consists of a series of EZAPPDPF macros (one macro for each table entry), followed by an Assembler END statement.

EZAPPDPF

You can code the EZAPPDPF macro to define default page formats instead of using the SCS (SNA Character Stream) printer emulation's default page format.

Syntax



Parameters

entryname
A required parameter specifying the user-defined name for this page format table entry. Each *entryname* must be unique and must conform to the rules for labels in assembler language. This name correlates to what is specified as the DPFNAME keyword in the ROUTING function of the EZAPPFL macro or as the DEF PAGE FORMAT field on the ISPF panel. See Figure 27 on page 79.

MPP=*value*
Represents the maximum presentation position or line length. This can be zero or any integer from 1 through 255, inclusive. If this parameter is omitted or explicitly set to zero, Network Print Facility interprets this as though MPP=80 was coded.

LM=*value*
Identifies where the left margin starts. This can be zero or any integer from 1 through MPP, inclusive. If this parameter is omitted or explicitly set to zero, Network Print Facility interprets this as column 1.

RM=*value*
Identifies where the right margin starts. This can be zero or any value from LM to MPP, inclusive. If this parameter is omitted or explicitly set to zero, Network Print Facility interprets this as though RM had been set equal to the MPP value.

HT=*value*
Identifies positions of horizontal tabs. Specify this as a list HT=(t1,t2,...,tn). Each tab can be zero or any value from LM to RM, inclusive. A value of zero is valid and ignored. The application data stream can add additional tab stops but cannot remove default tab stops.

MPL=*value*

Represents the maximum presentation line (page length) in lines. This can be zero or any integer from 1 through 255, inclusive. If this parameter is omitted or explicitly set to zero, Network Print Facility interprets this as 1 line.

TM=*value*

Identifies where the top margin starts. The top margin is also used as the line number for Select Vertical Channel 1. This can be zero or any value from 1 to MPL, inclusive. If this parameter is omitted or explicitly set to zero, Network Print Facility interprets this as line 1.

BM=*value*

Identifies where the bottom margin starts. This can be zero or any value from TM to MPL, inclusive. If this parameter is omitted or explicitly set to zero, Network Print Facility interprets this as though BM had been set equal to the MPL value. A bottom margin of 1 suppresses automatic form feed when the application spaces past the bottom margin.

VT=*value*

Identifies the position of vertical tabs (also channels 2–12). Specify as a list VT=(*t1,t2,...,tn*). The first 11 vertical tabs are also used as the line numbers for Select Vertical Channel 2 through 12. Each tab must be either zero or any value from TM to BM, inclusive. A value of zero is valid and ignored. The application data stream can add additional tab stops but cannot remove default tab stops.

STB.

The STB (suppress trailing blanks0 parameter indicates if trailing blanks should be suppressed (STB=YES) or kept (STB=NO). STB=YES is the default and allows blanks (x'40') to be removed from the end of each print record for LU1 sessions. Because this option reduces the amount of network data traffic, it is the NPF method used prior to APAR PN87720.

If NPF is being used to transmit data to a device other than a printer and the x'40' data value represents something other than an empty print position, specify STB=NO and the trailing blanks will be sent.

Usage notes

- If MPP or RM is set lower than the actual LPD line length, the full width is not used. If they are set greater than the actual LPD line length, then LPD either truncates the long lines (losing data) or splits the lines and disrupts the vertical spacing of the file.
- If BM is set to 1, NPF inserts no automatic form feeds. Explicit form feed or Select Vertical Channel 1 in the data stream cause form feeds. If BM is greater than 1, NPF moves lines that might print below BM to TM on the next page.
- The Network Print Facility assumes LPD starts each file at the top of a new page. The Network Print Facility also assumes that form feed causes LPD to position at line 1 of the next page, so the Network Print Facility inserts blank lines at the top of each page whenever the top margin is greater than 1. MPL should be the total number of print lines on the page. For correct formatting, the number of printable lines ((BM + 1) -TM) needs to be correct. If this number is too small, the full page length is not used. If it is too large, LPD prints over the page or inserts additional form feeds.
- The SCS presentation position command permits printing outside of the margins but within the physical page dimensions. If your application is using SCS PP, then all margin settings must be consistent with LPD.

Page format table examples

Figure 5 on page 36 shows what a user-modified page format table might look like. This does not resemble the IBM-supplied source code.

```
***** This first macro starts the  EZAPPPFT CSECT.
***** It uses all default values.
***** Bottom margin at line 1 suppresses automatic
***** form feed insertion.
 DEFAULT  EZAPPDPF
 *
 *
***** The next macro specifies
***** format for printers that support 80 columns
***** by 66 lines and places print data in columns
***** 1 through 80 and lines 1 through 66.
***** Set no horizontal tabs.  Set no vertical tabs.
***** Set CH01=1.  (CH02 through CH12 are not set.)
 LETTER   EZAPPDPF MPP=80,MPL=66       physical page columns, lines
 *
 *
***** The next macro specifies format for printers
***** that support 132 columns by 66 lines and places print
***** data in columns 10 through 120 and lines 6 through 60.
***** Set horizontal tabs in columns (10,) 25, 50, 75 and 100.
***** Set vertical tabs in lines (6,) 20, 30, 40 and 50.
***** Set CH01=6, CH02=20, CH03=30, CH04=40, and CH05=50.
*****     (CH06-CH12 are not set.)
WIDE     EZAPPDPF MPP=132,MPL=66,        physical page columns, lines  X
               LM=10,RM=120,TM=6,BM=60,  page margins                  X
               HT=(25,50,75,100),        horizontal tab stops          X
               VT=(20,30,40,50)          vertical tab stops
 *
 *
***** The next macro specifies format for printers
***** that support 255 columns by 255 lines.
***** Place print data in columns 15 through 250,
***** lines 10 through 250.
***** Set horizontal tabs in columns (15,) 25, 75 and 100.
***** Set vertical tabs in lines (10,) 15, 50, 100, 150, 250.
***** Set CH01=10, CH02=15, CH05=50, CH08=200, CH10=150,
***** and CH12=250.
***** (CH03, CH04, CH06, CH07, CH09, and CH11 are not set.)
MAX      EZAPPDPF MPP=255,MPL=255,        physical page columns, lines X
               LM=15,RM=250,TM=10,BM=250, page margins                 X
               HT=(25,100,0,75),          horizontal tab stops         X
               VT=(15,0,0,50,0,0,200,0,150,0,250)  vertical tab stops
 *
 *
          END   ,                  indicates end of module EZAPPPFT ****
```

Figure 5. Example of EZAPPDPF macro use

Chapter 4. Understanding the Network Print Facility files

Three files (VSAM key-sequenced data sets) serve as the foundation of the Network Print Facility processing. The ***routing*** and ***options*** files contain user-defined records that provide the Network Print Facility with the information it needs to route print data. The routing file defines where output will be printed, and the options file defines how it will print. The ***queue*** file contains temporary records created by the system to keep track of NPF print data sets.

In addition to these three files, NPF also maintains capture point and queue manager ***log*** files to track print data processing and a ***trace*** file to track NPF ISPF processing.

This section explains the type of information in each file and how the files relate to each other. You can find instructions for initializing and loading the NPF files in Chapter 5, "Creating the Network Print Facility files," on page 43 and for updating them in Chapter 6, "Maintaining the Network Print Facility files with ISPF," on page 59.

Note: The Network Print Facility maintains the integrity of the routing, options, and queue files in a shared file environment. Within a single MVS host, VSAM files can have shared access among the ISPF panels, an active VTAM capture point, an active FSS writer, and an active queue manager. The integrity mechanism assumes the files are allocated with at least SHAREOPTIONS (3,3) and will not perform correctly if less restrictive VSAM share options are specified. Users are strongly encouraged to use SHAREOPTIONS (4,3) to maintain data integrity.

The options file

An installation's print options are defined in the options file. Each record in the options file contains a set of valid LPR print options that are used when a job is printed. These options describe printers, type of data being printed, and instructions for printing the data. The LPR options can give instructions for a wide range of functions, such as whether or not to print headers, include margins, notify the sender, or translate the data. For specific information about the valid print options, see the LPR command in the z/OS Communications Server: IP User's Guide and Commands.

The Network Print Facility uses the options file with the routing file to determine ***how*** output will be printed. The routing file records contain pointers to options file records and thereby connect a set of print options to each routing. More than one routing file record can point to the same options file record.

The options file can also contain the name of a user-defined input record exit. This exit allows an installation to modify print data to meet specific output requirements. It can provide data manipulation capabilities beyond those provided by the LPR options, such as adding a banner page, modifying data by encryption, or deleting data. Use of any exit should not be done casually because it enables you to completely alter the data. See Chapter 7, "Writing exit routines to tailor the Network Print Facility," on page 93 for more information.

Working with the options file

To work with the Network Print Facility's options file, first allocate space using the IDCAMS utility with the specifications shown in Figure 6 on page 38.

Be sure to use the RECORDSIZE, SHAREOPTIONS, and KEYS values as defined. The size of the options file will be stable. Adjust FREESPACE based on experience with the files.

```
DEFINE CLUSTER (NAME(TCPIP.OPTIONS) VOLUMES(WRKLB2) -
     CYLINDERS(1 1)   -
     IMBED -
     RECORDSIZE(100 297) FREESPACE(0 15) -
     INDEXED SHAREOPTIONS(4 3) -
     UNIQUE NOWRITECHECK) -
     DATA ( -
      NAME(TCPIP.OPTIONS.DATA) -
      KEYS(16 0) ) -
     INDEX ( -
       NAME(TCPIP.OPTIONS.INDEX) )
```

Figure 6. Sample of options file allocation

After the space has been allocated, you can initialize and load records into the options file using the EZAPPFL macro. You can also use the ISPF interface to load options file records or to maintain and correct them. The interface includes panels for you to add, copy, update or delete options file records.

Options file input fields

When creating or updating an options file record, you need to enter data for the following fields.

Table 4. Option file record input fields

ISPF Field	**Macro Field**	**Definition**
OPTIONS NAME	OPTNAME	A unique alphanumeric field containing the name of the options record.
INPUT RECORD EXIT	IREXIT	The name of the input record exit, if one is used.
LPR OPTIONS	OPTIONS	A list of the LPR options.

See "Running the EZAPPFL macro" on page 44 and "Maintaining the options file" on page 69 for detailed instructions.

The routing file

The Network Print Facility uses a routing file or table to determine ***where*** to route output that will be processed on remote printers. The routing file is a VSAM key-sequenced data set that contains information to route each data set to its LPD printer queue. You need one routing file entry (VSAM record) for each printer being used at the installation. Each routing file record is identified with a unique key field consisting of a major name and a minor name. For JES printers, the JCL DEST, CLASS, and FORMS parameters are used for the routing file key fields. For VTAM printers, LU name of the logical printer is used.

The routing file records contain a pointer to options file records. Multiple routing file records can point to the same record in the options file. Because the same destination printer can be used to print output with different characteristics, it can have many entries in the routing file. Each variation will require a separate routing file record pointing to the appropriate LPR options.

The routing file has two types of routings: normal and specific broadcast:

- Normal routings have one destination and send output to one printer.
- Specific broadcast routings have more than one destination andsend output to multiple homogeneous printers. These printers of identical types (similar printers with different locations or destinations) have the same LPR options defined in the options file.

You can use the ISPF interface to modify the routing file when changes are required in your network, such as when printers are added or removed. This can be very useful if the network or a workstation stops functioning and the network administrator wants to reroute printer output.

Optionally, instead of altering the routing file, you can change the routing mechanism with a user exit. Use of any exit should not be done casually because it enables you to completely alter the data. See "Writing the general routing exit" on page 94 and "Writing the specific routing exit" on page 97 for more information on user-defined routing.

Working with the routing file

To work with the Network Print Facility's routing file, first allocate space using the IDCAMS utility with the specifications shown in Figure 7 on page 39. See the *VSAM Administration Guide* if you need more information about allocating VSAM data sets.

Be sure to use the RECORDSIZE, SHAREOPTIONS, and KEYS values as defined. The size of the routing file will be stable. Adjust FREESPACE based on experience with the files.

```
    DEFINE CLUSTER (NAME(TCPIP.ROUTING) VOLUMES(WRKLB2) -
         CYLINDERS(1 1)   -
         IMBED -
         RECORDSIZE(150 636) FREESPACE(0 15) -
         INDEXED SHAREOPTIONS(4 3) -
         UNIQUE NOWRITECHECK) -
         DATA ( -
          NAME(TCPIP.ROUTING.DATA) -
          KEYS(20 0) ) -
         INDEX ( -
           NAME(TCPIP.ROUTING.INDEX) )
```

Figure 7. Sample of routing file allocation

After the space has been allocated, you can initialize and load records into the routing file using the EZAPPFL macro. You can also use the ISPF interface to load routing file records or to maintain and correct them. The interface includes panels for you to add, copy, update or delete routing file records.

Routing file input fields

When creating or updating a routing file record, you need to input data for the following fields.

Table 5. Routing file record input fields

ISPF Field	Macro Field
MAJKEY	An 8-byte alphanumeric field containing the major name for this routing. For JES, this corresponds to the DEST parameter in the JCL. For VTAM, this is the logical printer LU name.
MINKEY	An 8-byte alphanumeric field containing the minor name for this routing. For JES, this corresponds to the CLASS and FORMS parameters in the JCL. For VTAM, this is any value of your choice.
NDEST	The number of destinations in this routing. The number 1 indicates a normal route to one printer; a number greater than 1 indicates a specific broadcast to a set of homogenous printers.
RETAINS	The period of time the system should retain data for this routing after a successful transmission.
RETAINU	The period of time the system should retain data for this routing after an unsuccessful transmission, after the retry attempts are exhausted.
RETRYT	The period of time the system should wait before attempting a retry on this routing after an unsuccessful attempt to transmit.
RETRYL.	The number of retry attempts to be made on this routing.
OPTNAME	The name of the options file record to be associated with this routing.
SREXIT	The name of a specific routing exit to be invoked.

Table 5. Routing file record input fields (continued)

ISPF Field	Macro Field
INAME	The internet name or IP address of the destination host for the first or only destination. This field is case-sensitive.
PNAME	The name of the printer to be used at the remote host for the first or only destination. This field is case-sensitive.
LUCLASS	For VTAM only, the VTAM classes to which this LU belongs.
DPFNAME	Foe VTAM only, the name of an entry in the page format table.
EOFNAME	For VTAM only, the name of an entry in the end-of-file rules table.

See "Running the EZAPPFL macro" on page 44 and "EZAPPFL TYPE=ROUTING" on page 45 for detailed instructions.

The queue file

For each print job, a Network Print Facility capture point application creates a temporary QSAM data set that contains the print data. It also adds a record to the queue file to keep track of the temporary print data set.

The queue file is a VSAM key-sequenced data set with one record for each print job. The key of a queue file record is based on the major name, minor name, and the name of the temporary QSAM print data set. The queue file record also contains status fields plus the retry time, retain times, and retry limit found in the routing file record for that destination.

After a queue file record has been created, it comes under the control of the queue manager. Through the use of the NPF ISPF interface, you can affect how the queue manager handles the queue file and the temporary print data sets.

- The NPF queue manager scans the queue file at regular intervals to determine what processing is needed for each record. It can make initial send attempts, retry failed transmissions and update the queue file records to reflect this activity, hold print data sets and queue file records, or delete print data sets and queue file records that are no longer needed.
- A user can modify the queue file records through the ISPF interface. The interface allows you to change the data set status, the next date and time to retry the transmission, the retain times for successful and unsuccessful transmissions, the internet and printer name, and the LPR options. These changes will affect whether the print data sets are sent, held, or deleted and where they are sent.

  It is possible for multiple NPF capture point applications to add records to the same queue file. However, only one NPF queue manager application should process any given queue file.

For a description of queue management, see "Managing the print queues" on page 85.

Working with the queue file

The only task required to set up the queue file is to allocate space using the IDCAMS utility with the specifications shown in Figure 8 on page 41. See the *VSAM Administration Guide* if you need more information about allocating VSAM data sets.

Be sure to use the RECORDSIZE, SHAREOPTIONS, and KEYS values as defined. The size of the queue file will be volatile. Adjust FREESPACE based on experience with the files.

```
DEFINE CLUSTER (NAME(TCPIP.QUEUE) VOLUMES(WRKLB2) -
     CYLINDERS(1 1)    -
     IMBED -
     RECORDSIZE(870 1000) FREESPACE(0 15) -
     INDEXED SHAREOPTIONS(4 3) -
     UNIQUE NOWRITECHECK) -
     DATA ( -
      NAME(TCPIP.QUEUE.DATA) -
      KEYS(62 0) ) -
     INDEX ( -
      NAME(TCPIP.QUEUE.INDEX) )
```

Figure 8. Sample of queue file allocation

After the space has been allocated, the NPF capture point applications will create queue file records and the NPF queue manager will delete them at the appropriate time.

The log file

The Network Print Facility has a logging facility that writes messages in a log file to track the system activity regarding temporary print data sets. The messages record when a print data set is created, deleted, placed on the queue, and successfully or unsuccessfully transmitted. See "Network Print Facility logging" on page 127 for more information on these messages.

The log file data set has sequential organization (PS), a fixed block format (FBA), a logical record length (LRECL) of 133, and a block size (BLKSIZE) of 1330. Because the log file is not a VSAM data set, you do not need to allocate space for it. It can be dynamically allocated by specifying the log file with the //EZAPPLOG DD statement in the JCL to start the capture point applications and queue manager.

The log file cannot be shared. Therefore, you will need a unique log file for each startup procedure.

If the //EZAPPLOG DD statement is missing, the capture point applications will run but the system will not log the activity.

The SNAP File for the VTAM capture point

The NPF VTAM capture point optionally writes debugging information to the SNAP DD. Do not use this DD card unless requested by IBM service. The //EZAPPSNP DD can use SYSOUT=* or a QSAM data set with VBA or FBA format, a logical record length (LRECL) of 133, and any blocksize.

The SNAP file is specified with the //EZAPPSNP DD statement in the JCL to start the capture point. The SNAP file cannot be shared; therefore, you must specify a unique SNAP file for each startup procedure. If the //EZAPPSNP statement is missing, the system does not perform the debug SNAP trace.

The trace file

The NPF system also has a tracing facility to track the NPF internal processing that occurs during interactive ISPF sessions. The tracing facility can be turned on and off from the interface. It is normally used only for short periods of time when diagnosing problems and collecting information to forward to IBM service representatives.

Your system can have as many trace files as needed. The trace file that is used during an ISPF session is specified on the NPF Diagnosis Functions Panel. If a user specifies a data set that does not exist and the data set name begins with the user's TSO user ID, the system will dynamically allocate the file. All other trace files (those with a high-level qualifier other than the user ID) must be allocated with the following specifications: sequential organization (PS), a variable block format (VB), a logical record length (LRECL) of 132, and any block size (BLKSIZE).

Chapter 5. Creating the Network Print Facility files

The Network Print Facility provides a macro interface, EZAPPFL, to initialize the routing, options, and queue files. The program generated by the macro allows you several choices to initialize and load the files.

- Method 1: Initialize all three files. The macro loads each of them with a single record of binary zeroes.
- Method 2: Initialize all three files and add routing file and options file records at the same time. The program generated by the macro loads the queue file with a single record of binary zeroes and loads the routing and options files with the records you define.
- Method 3: Initialize the three files using either method 1 or 2 and then run the macro in update mode at another time to add records to the routing file and options file. When you run the macro in this mode, it appends records to existing files.

After the files have been initialized and loaded, you can use the ISPF interface to make additions or corrections to the routing and options files. You can choose to add all your routing and options records using the ISPF interface or just use it to make minor updates. In any case, you ***must*** run the EZAPPFL macro first to initialize and load the files.

Generally initializing and loading procedures

Before you use the ISPF interface to make additions or corrections to the routing and options files, you need to initialize and load the files first. The NPF VSAM files can be initialized only by using the EZAPPFL macro in a batch process.

Procedure

Perform these following steps to initiate the files.

1. Define the files using IDCAMS.

   A sample that you can copy and modify is provided with the installation. See "Joint allocation of NPF VSAM files" on page 53.

   Make note of the data set names you specify in this step and use them in the JCL for all other NPF jobs.
2. Code the EZAPPFL macro to define the load program. Use the macro with TYPE=INITIAL and MODE=LOAD and then issue a second macro with TYPE=FINAL to initialize the records. Optionally, use the macro with TYPE=ROUTING, TYPE=NXTDEST, and TYPE=OPTIONS to load routing and options records. See the syntax descriptions and examples in "Running the EZAPPFL macro" on page 44.
3. Run a job stream to assemble the macros and link and run the load program.

   See the sample shown in "Initialization of the NPF VSAM files" on page 54.

   The program uses the following ddnames. These will be consistent in the JCL for all other NPF jobs.

ddname	**File**
EZAPPRF	Routing
EZAPPOF	Options
EZAPPQF	Queue

Results

After the files have been initialized, you can add records to the options and routing files using the EZAPPFL macro or the NPF ISPF interface. The queue file is under system control and you cannot add records to it. The NPF ISPF interface allows you to view, modify, and delete queue file records so that you can administer your print queues.

There are two steps to the EZAPPFL macro update process:

1. Code the EZAPPFL macro to define the update program. Use TYPE=INITIAL with MODE=UPDATE, TYPE=ROUTING, TYPE=NXTDEST, TYPE=OPTIONS and TYPE=FINAL.
2. Run a job stream to assemble the macros, and then link and run the update program. See the sample shown in "Adding records to the NPF VSAM files" on page 56.

To interactively update the routing and options files and manage the print queues, see Chapter 6, "Maintaining the Network Print Facility files with ISPF," on page 59.

Running the EZAPPFL macro

The EZAPPFL macro has five TYPE specifications that perform different functions. The syntax for each macro type is described separately.

TYPE=INITIAL
Defines the beginning of the program creation process and is required. You can specify MODE=LOAD to initialize and load the files or MODE=UPDATE to add records to the routing and options files.

TYPE=ROUTING
Defines a normal or specific broadcast routing. If routings are defined, they must be specified in the ascending alphanumeric sequence of the major and minor keys, and they must precede any options record definitions.

TYPE=NXTDEST
Defines an additional destination for a previous routing. The use of the NXTDEST type indicates a specific broadcast routing. When the number of destinations is greater than 1, the NXTDEST type must be used to define all of the destinations except the first (which is defined with the EZAPPFL TYPE=ROUTING macro). This indicates the addition of specific destinations.

TYPE=OPTIONS
Defines an options record. If options records are defined, they must be specified in ascending alphanumeric sequence (according to the OPTNAME parameter).

TYPE=FINAL
Defines the end of the program creation process and is required.

These examples show when and in what order you would code the various the EZAPPFL types. Macro types must be coded in the order shown, however, the other macro parameters within a specific type can be coded in any order.

Note: These examples are not complete. They do not include all the parameters needed to code a complete macro and the continuation indicator in column 72 required by the assembler program is omitted.

- To define a program that creates the binary zero records for all files:

```
         EZAPPFL TYPE=INITIAL
         EZAPPFL TYPE=FINAL
```

- To define a program that creates 1 routing record and 1 options record:

```
         EZAPPFL TYPE=INITIAL
         EZAPPFL TYPE=ROUTING
         EZAPPFL TYPE=OPTIONS
         EZAPPFL TYPE=FINAL
```

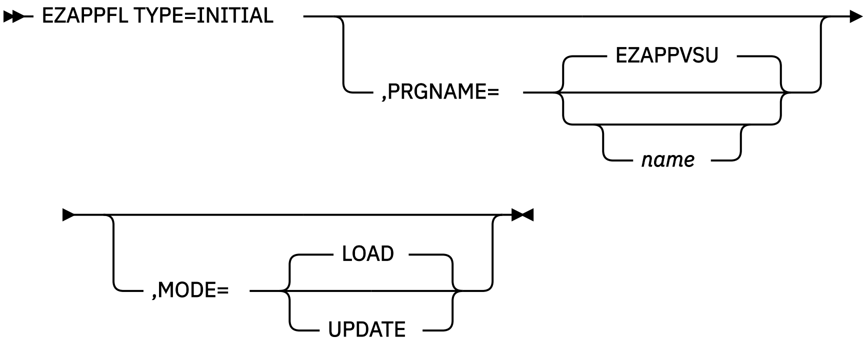
- To define a program that creates 1 routing record with 4 destinations and 1 options record:

```
          EZAPPFL TYPE=INITIAL
          EZAPPFL TYPE=ROUTING
          EZAPPFL TYPE=NXTDEST
          EZAPPFL TYPE=NXTDEST
          EZAPPFL TYPE=NXTDEST
          EZAPPFL TYPE=OPTIONS
          EZAPPFL TYPE=FINAL
```

EZAPPFL TYPE=INITIAL

The INITIAL function indicates the start of the macro definition. It is required.

Syntax



Parameters

PRGNAME=EZAPPVSU|*name*
Optional 8-character name assigned to the program being created. The default name is EZAPPVSU.

MODE=LOAD|UPDATE
Optional mode of operation.

LOAD
Specifies new files are being created. This is the default.

UPDATE
Specifies the routing and options files are being appended.

Examples

This example shows how to code the initial load.

```
          EZAPPFL TYPE=INITIAL,                                         X
                PRGNAME=INITLOAD,                                       X
                MODE=LOAD
```

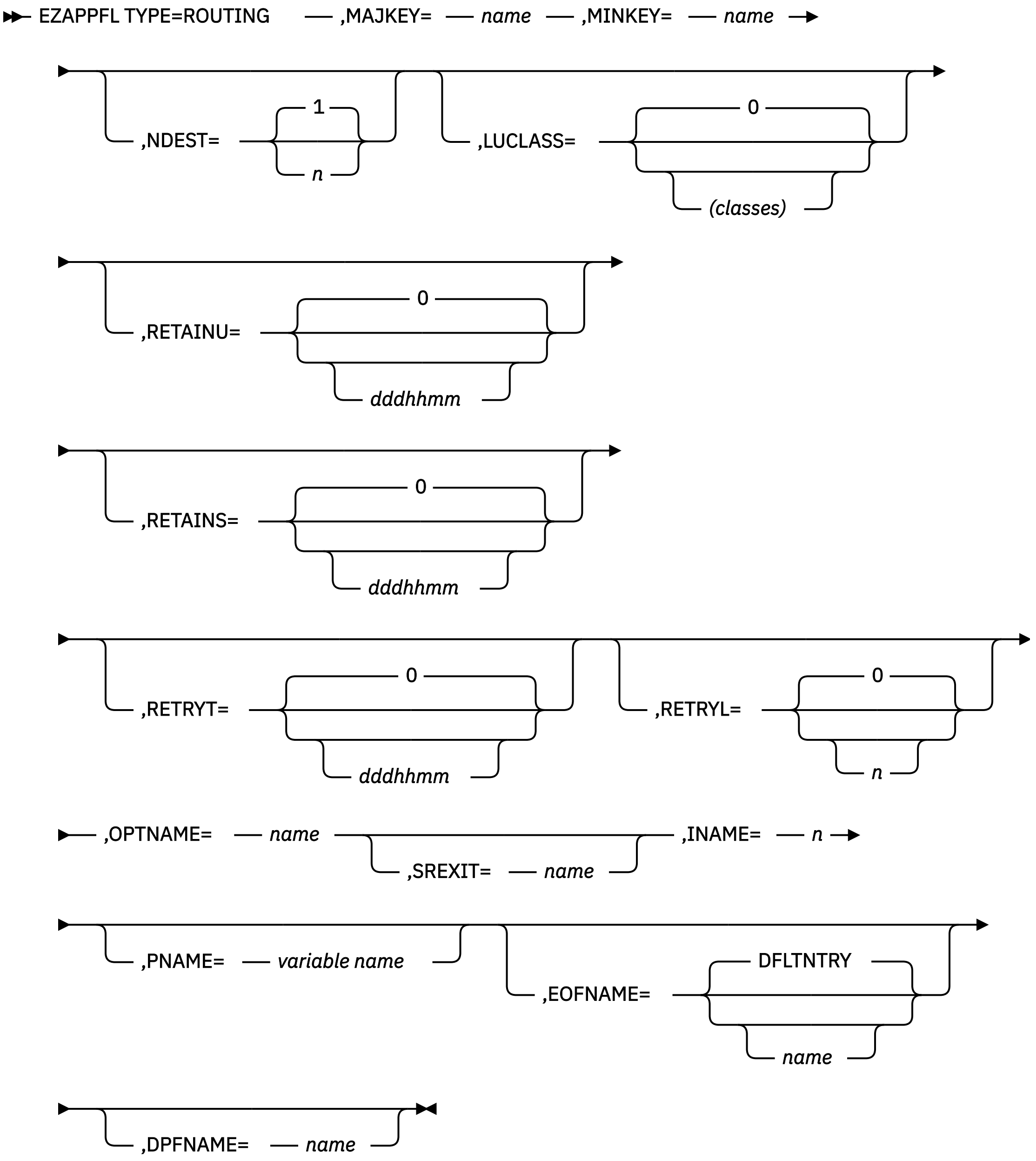
This example shows how to code an update to the routing and options files.

```
          EZAPPFL TYPE=INITIAL,                                         X
                PRGNAME=ADDRECS,                                        X
                MODE=UPDATE
```

EZAPPFL TYPE=ROUTING

The ROUTING function defines either a normal or specific broadcast routing. If routing records are defined, they must be specified in ascending alphanumeric order according to the sequence of the major and minor keys.

Syntax



Parameters

MAJKEY=*name*

A 1 to 8 alphanumeric character description of the major key for the routing. For JES, this corresponds to the DEST parameter in the JCL. For VTAM, this is the logical printer LU name.

This required parameter corresponds to the MAJOR NAME value used in the ISPF panel interface. (See Figure 26 on page 78.)

MINKEY=*name*

A 1 to 8 alphanumeric character description of the minor key for the routing. For JES, this corresponds to 1 character for the CLASS and 4 characters or more for the FORMS parameters in the JCL. For VTAM, this parameter must be filled in because it is used for internal file access. It can contain any 1 to 8 alphanumeric characters of your choice but the first character must be alphabetic.

This required parameter corresponds to the MINOR NAME value used in the ISPF panel interface. (See Figure 26 on page 78.)

NDEST=1|*n*

A 2 byte numeric field specifying the number of destinations in this routing. More than one destination implies a specific broadcast routing to send output to multiple homogeneous printers. If the NDEST value is greater than 1, it requires an NXTDEST function of the EZAPPFL macro to specify the IP address and printer name for ***each*** of the other destinations. For example, a broadcast with 3 destinations requires one TYPE=ROUTING with NDEST=3 followed by two TYPE=NXTDEST macros.

If you omit NDEST, the system supplies a default of 1. This parameter corresponds to the NO OF DEST field in the NPF ISPF interface (see Figure 27 on page 79).

LUCLASS=0|(n1,n2,.....)

The VTAM classes to which this LU belongs. Valid values are 1 - 64. The assigned classes are compared to the classes specified in the VTAM startup to determine which LUs are to be managed by a VTAM capture point. LUCLASS is used only to initialize a grouping of LUs. Parentheses are required to enclose one or more classes. Classes are separated by commas. This field corresponds to the field of the same name on the NPF ISPF interface (see Figure 27 on page 79).

LUCLASS is optional, the default is 0 (indicating a JES route.)

Do not specify this parameter for JES; for JES it is ignored.

RETAINU=0|*dddhhmm*

The amount of time an unsuccessfully transmitted print file is retained for this routing after completing the specified number of retries. The format is *dddhhmm* where:

ddd

The number of days to retain the file (range 000-366)

hh

The number of hours to retain the file (range 00-23)

mm

The number of minutes to retain the file (range 00-59)

The total time to retain the file is the sum of the days, hours, and minutes. This field corresponds to the RETAIN TIME (U) field on the NPF ISPF interface (see Figure 27 on page 79).

This is optional, the default is 0.

RETAINS=0|*dddhhmm*

The amount of time a successfully transmitted print file is retained for this routing. The format is *dddhhmm* where:

ddd

The number of days to retain the file (range 000-366)

hh

The number of hours to retain the file (range 00-23)

mm

The number of minutes to retain the file (range 00-59)

The total time to retain the file is the sum of the days, hours, and minutes.

This field corresponds to the RETAIN TIME (S) field on the NPF ISPF interface (see Figure 27 on page 79).

This is optional, the default is 0.

RETRYT=0|*dddhhmm*

The retry time for this routing is the time between retries for a print file that has not been successfully sent. This field has the format *dddhhmm* where:

ddd

The number of days to wait before a retry (range 000-366)

hh

The number of hours to wait before a retry (range 00-23)

mm

The number of minutes to wait before a retry (range 00-59)

The total time to wait is the sum of the days, hours, and minutes. The maximum time allowed is 1 year. This field corresponds to the RETRY INTERVAL field on the NPF ISPF interface (see Figure 27 on page 79).

This is optional; the default is 0.The queue manager will treat the default of 0 as a 30-second interval.

RETRYL=0|*n*

The maximum number of retries for a print file. The maximum limit is 32 768. This field corresponds to the RETRY LIMIT field on the NPF ISPF interface (see Figure 27 on page 79).

This is optional, the default is 0.

OPTNAME=*name*

The 1–16 alphanumeric name of the options record. This is the name referred to in the OPTIONS function of the EZAPPFL macro by the OPTNAME parameter. It must be unique and can have up to 16 alphanumeric characters (A–Z and 0–9). If the name is not 16 characters, it is padded to the right with blanks.

This field corresponds to the OPTIONS NAME field on the NPF ISPF interface (see Figure 27 on page 79).

This is a required parameter.

SREXIT=*name*

The 1 to 8 alphanumeric character name of the specific routing exit associated with this routing. If the name is not 8 characters, it is padded to the right with blanks. This field corresponds to the ROUTING EXIT field on the NPF ISPF interface (see Figure 27 on page 79).

This parameter is optional, the default is no exit.

INAME=n

The internet name or IP address of this destination. This required parameter can be up to 255 characters in length, is case-sensitive, and corresponds to the HOST NAME/IP ADDRESS field on the NPF ISPF interface (see Figure 27 on page 79).

Be sure to specify the host in the same way on all routing records which specify the same printer. This is necessary to ensure correct FIFO (First In, First Out) handling of print jobs for that printer.

PNAME=*name*

The printer name for this destination. This required parameter can be up to 255 characters in length, is case-sensitive, and corresponds to the PRINTER NAME field on the NPF ISPF interface (see Figure 27 on page 79).

DPFNAME=*name*

For VTAM only, 1 - 8 byte alphanumeric name of an entry in the page format table. The default is blank. This corresponds to DEF PAGE FORMAT used in the ISPF panel interface. (See Figure 27 on page 79.) You can find information about the page format table in "Page format definition" on page 32.

For JES, this parameter is ignored.

EOFNAME=*name*

For VTAM only, 1 - 8 byte alphanumeric name of an entry in the end-of-file rules table. The default is DFLTNTRY. This corresponds to EOFILE NAME used in the ISPF panel interface. (See Figure 27 on page 79.) You can find information about the page format table in "End-of-File definition" on page 25.

For JES, this parameter is ignored.

Usage notes

- The first character of both the major and minor name must be alphabetic.
- Code TYPE=ROUTING macros in ascending alphanumeric order based on the MAJKEY and MINKEY parameters.
- All TYPE=ROUTING macros must follow the TYPE=INITIAL macro.
- All TYPE=ROUTING macros must precede any EZAPPFL TYPE=OPTIONS macros.
- If the NDEST value is greater than 1 (indicating a specific broadcast routing) and it is not followed by the correct number of TYPE=NXTDEST macros, the Network Print Facility does not process the routing and issues console error message EZY0633E. For example, if a TYPE=ROUTING macro with NDEST=2 is not followed by one TYPE=NXTDEST macro, the routing will not be processed.
- To avoid the loss of output data sets until you are certain that they have been successfully sent and printed, define values for the retain time (RETAINS and RETAINU), retry time (RETRYT), and retry limit (RETRYL) parameters. If none of these values are specified, a print data set sent to this routing will not be saved regardless of whether or not it was successfully transmitted.

  Factors to consider when coding these parameters:

  - The NPF FSS writer and NPF VTAM capture point applications store print jobs in temporary sequential data sets as they are received. The capture point applications pass control to the &halpp queue manager, which manages the sending of the print data to LPD. RETAINS, RETAINU, RETRYT, and RETRYL affect how the queue manager handles these temporary data sets.
  - RETRYL and RETRYT pertain to files that are not successfully delivered to their target LPD server. These determine how many times and how often the queue manager should attempt to send this file.

    Set RETRYT at a reasonable interval for network or LPD problems to clear. Set RETRYL to control how many times to try resend before assuming that the problem is persistent.
  - RETAINU and RETAINS specify a period of "administrative" time to allow files that are lost through LPD print queue failures, or by users, to be resent. They provide time for system administrators to examine failed jobs for problem determination and correction.

    RETAINS affects all data sets that are eligible for deletion after they were successfully sent. RETAINU affects all data sets that are eligible for deletion after they have exhausted their retry count.

Examples

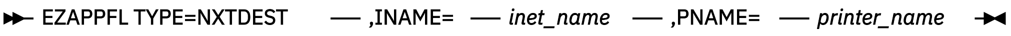
This example shows the coding for a normal routing for VTAM.

```
         EZAPPFL TYPE=ROUTING,                                         X
               MAJKEY=ELU10100,                                        X
               MINKEY=VTAM,                                            X
               RETAINS=5,                                              X
               RETAINU=10,                                             X
               RETRYT=5,                                               X
               RETRYL=2,                                               X
               NDEST=1,                                                X
               OPTNAME=OPTION2,                                        X
               LUCLASS=(1,8,16,32,35,64),                              X
               DPFNAME=PGFORMT1,                                       X
               EOFNAME=EOFPARM1,                                       X
               SREXIT=SPCEXIT2,                                        X
               INAME=MVS1.TCP.RALEIGH.IBM.COM,                         X
               PNAME=LPT1
```

EZAPPFL TYPE=NXTDEST

The NXTDEST function defines additional destinations on a specific broadcast routing. This routing must first have an EZAPPFL TYPE=ROUTING macro with the number of destinations greater than 1. Each additional destination in a broadcast routing requires a TYPE=NXTDEST entry.

Syntax



Parameters

INAME=*inet_name*
The internet name or IP address of this destination. This required parameter is case-sensitive.

PNAME=*printer_name*
The printer name for this destination. This required parameter is case-sensitive.

Usage notes

- All TYPE=NXTDEST macros must follow a TYPE=ROUTING macro that defines a specific broadcast routing.
- All TYPE=ROUTING and TYPE=NXTDEST macros must precede any TYPE=OPTIONS macros.
- The number of TYPE=NXTDEST macros must be 1 less than the value specified in the NDEST parameter in the preceding TYPE=ROUTING macro. If this is not true, the Network Print Facility does not process the routing and issues console error message EZY0633E. For example, if a TYPE=ROUTING macro with NDEST=2 is not followed by one TYPE=NXTDEST entry, the routing will not be processed.

Examples

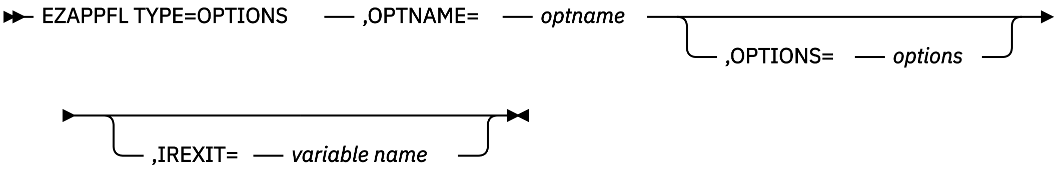
This example shows a specific broadcast routing with 2 destinations for the JES FSS writer.

```
         EZAPPFL TYPE=ROUTING,                                         X
               MAJKEY=LOCAL,                                           X
               MINKEY=C1185,                                           X
               RETAINS=5,                                              X
               RETAINU=10,                                             X
               RETRYT=5,                                               X
               RETRYL=2,                                               X
               NDEST=2,                                                X
               OPTNAME=OPTION1,                                        X
               SREXIT=SPECEXIT,                                        X
               INAME=9.67.112.25,                                      X
               PNAME=LPT1
*
         EZAPPFL TYPE=NXTDEST,                                         X
               INAME=9.67.112.34,                                      X
               PNAME=LPT2
```

EZAPPFL TYPE=OPTIONS

The OPTIONS function defines an options record. If options records are defined, they must be specified in ascending alphanumeric sequence.

Syntax



Parameters

OPTNAME=*optname*

The name of the options record. This is the name referred to in the EZAPPFL TYPE=ROUTING macro by the OPTNAME parameter. It must be unique and can have up to 16 alphanumeric characters (A–Z and 0–9). If the name is not 16 characters, it is padded to the right with blanks.

This parameter provides the key for the VSAM record and is required. It corresponds to the OPTIONS NAME field in the NPF ISPF interface. See Figure 18 on page 72.

OPTIONS=*options*

The list of LPR options associated with this routing. This field can have up to 253 characters. Multiple options must be separated by blanks and enclosed in single quotation marks. Values and defaults can be found in the z/OS Communications Server: IP User's Guide and Commands . The only requirement specific to NPF is that you provide the JOB parameter in full (not just the first letter J) and specify the jobname or use a comma in its place. For example, these are valid:

```
job sms11,pass=****,for=usr11b

job ,pass=****

JOB SMS11
```

These are not valid:

```
job pass=****,for=usr11b

JOB PASS=****

job
```

Any options you specify here override default LPR options. If you want the default set of options supplied by LPR, do not specify the OPTIONS parameter. Some options will always be provided by NPF, such as jobname and user name for JES, user name for VTAM.

This corresponds to the LPR OPTIONS field in the NPF ISPF interface. See Figure 18 on page 72.

IREXIT=*name*

Up to an 8-character name of the input record exit **if one is to be used.** This corresponds to the INPUT RECORD EXIT field in the NPF ISPF interface. See Figure 18 on page 72. This parameter is optional. The default is no exit.

Usage notes

- The input record exit can insert less than 208 characters into the LPR options field.
- LPR translates EBCDIC files to an ASCII data stream for LPD. LPR supports translation to several languages or a user-supplied translation. For translation to a particular language, specify the appropriate LPR translation option. For more about translation options, see the LPR options described in z/OS Communications Server: IP User's Guide and Commands and the using translation tables topic in z/OS Communications Server: IP Configuration Guide.

Examples

This example shows how to code one LPR option and an input exit.

```
         EZAPPFL TYPE=OPTIONS,                                        X
               OPTNAME=OPTION1,                                       X
               IREXIT=INRECEX1,                                       X
               OPTIONS=TRACE
```

This example shows how to code multiple LPR options.

```
         EZAPPFL TYPE=OPTIONS,                                        X
               OPTNAME=OPTION2,                                       X
               OPTIONS='burst header'
```

EZAPPFL TYPE=FINAL

The FINAL function defines the end of the program creation process and is required.

Syntax

```
►►─ EZAPPFL TYPE=FINAL ─►◄
```

Parameters

None

Required parameters and default values for EZAPPFL

The following table provides a summary of the required parameters and default values for the EZAPPFL macro interface:

Table 6. EZAPPFL macro parameters and defaults

Macro	Parameter	Required or Default
TYPE=INITIAL	PRGNAME	default=EZAPPVSU
	MODE	default=LOAD
TYPE=ROUTING	MAJKEY	required
	MINKEY	required
	NDEST	default=1
	LUCLASS	default=0
	RETAINS	default=0
	RETAINU	default=0
	RETRYT	default=0
	RETRYL	default=0
	OPTNAME	required
	SREXIT	default=blank (no exit)
	DPFNAME	default=blank
	EOFNAME	default=DFLTNTRY
TYPE=NXTDEST	INAME	required
	PNAME	required
TYPE=OPTIONS	OPTNAME	required
	IREXIT	default=blank (no exit)
	OPTIONS	default=blank (default LPR options)

Sample NPF initialization jobs

The following sample jobs can be found in *hlq*.SEZAINST. Copy them from this library and update them as needed to suit your configuration.

DELALLOC
Jointly allocates the routing, options, and queue files by deleting and defining the VSAM data sets

ROUTEOPT
Initializes and loads the routing, options, and queue files after they have been allocated

ADDRTOPT
Adds records to the routing and options files after they have been initialized.

Joint allocation of NPF VSAM files

You can copy this sample from *hlq*.SEZAINST(DELALLOC) and use it to jointly allocate the routing, options, and queue files by deleting and defining the VSAM data sets.

Execution note

You will need to modify the data set names as required by your installation.

```
//DELALLOC JOB
//* -------------------------------------------------------------------*
//*
//*  COPYRIGHT = NONE
//*
//*  SMP/E Distribution Name: EZAEC042                                 *
//*
//*  This is sample JCL to delete and allocate the VSAM data sets
//*  needed for the Network Print Facility.
//*
//*    To delete the VSAM data sets required by the Network Print
//*    Facility use step 1 of this JCL.
//*
//*    To allocate the VSAM data sets required by the Network Print
//*    Facility use step 2 of this JCL.
//*
//*    WARNING:
//*
//*       *  Do not use this JCL to delete a queue file containing
//*          records for one or more temporary print data sets.
//*          Without that queue file, there will be no way for the
//*          ISPF panel interface to list those temporary data sets
//*          or for the queue manager to manage their further
//*          processing.
//*
//* -------------------------------------------------------------------*
//*  Step 1 :
//*
//*    The VSAM data sets deleted in this step are :
//*
//*         "Routing file" data sets
//*                              TCPIP.ROUTING
//*                              TCPIP.ROUTING.DATA
//*                              TCPIP.ROUTING.INDEX
//*         "Options file" data sets
//*                              TCPIP.OPTIONS
//*                              TCPIP.OPTIONS.DATA
//*                              TCPIP.OPTIONS.INDEX
//*         "Queue file" data sets
//*                              TCPIP.QUEUE
//*                              TCPIP.QUEUE.DATA
//*                              TCPIP.QUEUE.INDEX
//*
//* -------------------------------------------------------------------*
//DEL      EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
       DELETE -
            TCPIP.ROUTING -
            PURGE -
            ERASE
       DELETE -
            TCPIP.OPTIONS -
            PURGE -
            ERASE
       DELETE -
            TCPIP.QUEUE -
            PURGE -
            ERASE
//* -------------------------------------------------------------------*
//*  Step 2 :
//*
//*    The VSAM data sets allocated in this step are :
//*
//*         "Routing file" data sets
//*                             TCPIP.ROUTING
//*                             TCPIP.ROUTING.DATA
//*                             TCPIP.ROUTING.INDEX
//*         "Options file" data sets
//*                             TCPIP.OPTIONS
//*                             TCPIP.OPTIONS.DATA
//*                             TCPIP.OPTIONS.INDEX
//*
//*         "Queue file" data sets
//*                             TCPIP.QUEUE
//*                             TCPIP.QUEUE.DATA
//*                             TCPIP.QUEUE.INDEX
//*
//* -------------------------------------------------------------------*
//CREATE   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
   DEFINE CLUSTER (NAME(TCPIP.ROUTING) VOLUMES(WRKLB2) -
        CYLINDERS(1 1)   -
        IMBED -
        RECORDSIZE(150 636) FREESPACE(0 15) -
        INDEXED SHAREOPTIONS(4 3) -
        UNIQUE NOWRITECHECK) -
        DATA ( -
          NAME(TCPIP.ROUTING.DATA) -
          KEYS(20 0) ) -
        INDEX ( -
          NAME(TCPIP.ROUTING.INDEX) )
   DEFINE CLUSTER (NAME(TCPIP.OPTIONS) VOLUMES(WRKLB2) -
        CYLINDERS(1 1)   -
        IMBED -
        RECORDSIZE(100 297) FREESPACE(0 15) -
        INDEXED SHAREOPTIONS(4 3) -
        UNIQUE NOWRITECHECK) -
        DATA ( -
          NAME(TCPIP.OPTIONS.DATA) -
          KEYS(16 0) ) -
        INDEX ( -
          NAME(TCPIP.OPTIONS.INDEX) )
   DEFINE CLUSTER (NAME(TCPIP.QUEUE) VOLUMES(WRKLB2) -
        CYLINDERS(1 1)   -
        IMBED -
        RECORDSIZE(870 1000) FREESPACE(0 15) -
        INDEXED SHAREOPTIONS(4 3) -
        UNIQUE NOWRITECHECK) -
        DATA ( -
          NAME(TCPIP.QUEUE.DATA) -
          KEYS(62 0) ) -
        INDEX ( -
          NAME(TCPIP.QUEUE.INDEX) )
```

Initialization of the NPF VSAM files

You can copy this sample from *hlq*.SEZAINST(ROUTEOPT) and use it to initialize and load the routing, options, and queue files after they have been allocated.

This job assembles the EZAPPFL macros and then links and executes the load program.

Execution notes

- You will need to include your own macro parameters to define your routing and options file records.
- You will need to modify the data set names specified by //EZAPPRF, //EZAPPOF, and //EZAPPQF, as required by your installation.

```
//ROUTEOPT JOB
//* -------------------------------------------------------------------*
//*                                                                    *
//*  COPYRIGHT = NONE                                                  *
//*                                                                    *
//*  SMP/E Distribution Name: EZAEC0YY                                 *
//*                                                                    *
//*  This is sample JCL to assemble, link and execute the file load    *
//*  program for the routing, options and queue data sets used by      *
//*  the Network Print Facility.                                       *
//*                                                                    *
//*    In this sample, the names of the routing, options and queue     *
//*    files are:                                                      *
//*                                                                    *
//*       TCPIP.ROUTING                                                *
//*       TCPIP.OPTIONS                                                *
//*       TCPIP.QUEUE                                                  *
//*                                                                    *
//*    All three are assumed to have been created and initialized      *
//*    prior to the use of this JCL.                                   *
//*                                                                    *
//* -------------------------------------------------------------------*
//*  Step 1 :                                                          *
//*                                                                    *
//*    This step assembles the File Load Program                       *
//*                                                                    *
//*    This is the area where you will enter the routing and options   *
//*    information.                                                    *
//* -------------------------------------------------------------------*
//ASM       EXEC PGM=IEV90,PARM='OBJECT,TERM',REGION=1024K
//SYSLIB    DD DSN=TCPIP.SEZACMAC,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSPUNCH  DD DUMMY
//SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(80,(500,50))
//SYSTERM   DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
* --------------------------------------------------------------------*
*     Must start with INITIAL parm                                    *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=INITIAL,                                         X
               PRGNAME=ROUTEOPT,                                       X
               MODE=LOAD
* --------------------------------------------------------------------*
*     Example of a normal route for VTAM                              *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=ROUTING,                                         X
               MAJKEY=ELU10100,                                        X
               MINKEY=VTAM,                                            X
               RETAINS=5,                                              X
               RETAINU=10,                                             X
               RETRYT=5,                                               X
               RETRYL=2,                                               X
               NDEST=1,                                                X
               OPTNAME=OPTION2,                                        X
               LUCLASS=(1,8,16,32,35,64),                              X
               DPFNAME=PGFORMT1,                                       X
               EOFNAME=EOFPARM1,                                       X
               INAME=MVS1.TCP.RALEIGH.IBM.COM,                         X
               PNAME=LPT1
* --------------------------------------------------------------------*
*     Example of a specific broadcast route for the JES writer        *
*     (with 2 destinations)                                           *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=ROUTING,                                         X
               MAJKEY=LOCAL,                                           X
               MINKEY=C1185,                                           X
               RETAINS=5,                                              X
               RETAINU=10,                                             X
               RETRYT=5,                                               X
               RETRYL=2,                                               X
               NDEST=2,                                                X
               OPTNAME=OPTION1,                                        X
               INAME=9.67.112.25,                                      X
               PNAME=LPT1
* --------------------------------------------------------------------*
*     Example of second destination for the specific broadcast        *
*     route defined above                                             *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=NXTDEST,                                         X
               INAME=QUIDDITY,                                         X
               PNAME=LPT2
* --------------------------------------------------------------------*
*     Examples of options entries                                     *
* --------------------------------------------------------------------*
*     OPTION1                                                          *
* ---------------------------------------------------------------------*
         EZAPPFL TYPE=OPTIONS,                                          X
               OPTNAME=OPTION1,                                         X
               IREXIT=INRECEX1,                                         X
               OPTIONS=TRACE
* ---------------------------------------------------------------------*
*     OPTION2                                                          *
* ---------------------------------------------------------------------*
         EZAPPFL TYPE=OPTIONS,                                          X
               OPTNAME=OPTION2,                                         X
               OPTIONS='TRACE CLASS E'
* ---------------------------------------------------------------------*
*     Must end with FINAL parm                                         *
* ---------------------------------------------------------------------*
         EZAPPFL TYPE=FINAL
         END
/*
//* -------------------------------------------------------------------*
//*  Step 2 :                                                           *
//*                                                                     *
//*    This step links the File Load Program                           *
//* -------------------------------------------------------------------*
//LKED    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
//             REGION=512K,COND=(4,LT)
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
//SYSLMOD   DD DSNAME=&&GOSET(GO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,1,1))
//SYSLIN    DD DSNAME=&&OBJSET,DISP=(OLD,DELETE)
//LINK.SYSIN  DD DUMMY
//* --------------------------------------------------------------------*
//*  Step 3 :                                                           *
//*                                                                     *
//*    This step executes the File Load Program                         *
//* --------------------------------------------------------------------*
//LOAD     EXEC PGM=*.LKED.SYSLMOD,REGION=512K,COND=(4,LT)
//EZAPPRF  DD   DSN=TCPIP.ROUTING,DISP=OLD
//EZAPPOF  DD   DSN=TCPIP.OPTIONS,DISP=OLD
//EZAPPQF  DD   DSN=TCPIP.QUEUE,DISP=OLD
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSMDUMP DD   SYSOUT=*
```

Adding records to the NPF VSAM files

You can copy this sample from *hlq*.SEZAINST(ADDRTOPT) and use it to add (append) routing and options records after the files have been allocated and initialized. You can append records if the routing and options files were loaded with binary zero records or with actual data.

This job assembles the EZAPPFL macros and then links and executes the load program.

Execution notes

- You will need to include your own macro parameters to define your routing and options file records.
- You will need to modify the data set names specified by //EZAPPRF, //EZAPPOF, and //EZAPPQF as required by your installation.

```
//ADDRTOPT  JOB
//* --------------------------------------------------------------------*
//*                                                                     *
//*  COPYRIGHT = NONE                                                   *
//*                                                                     *
//*  SMP/E Distribution Name: EZAEC0YR                                  *
//*                                                                     *
//*  This is sample JCL to add routing and options records to           *
//*  existing Network Print Facility VSAM data sets.                    *
//*                                                                     *
//*  In this sample, the names of the routing, options and queue        *
//*  files are:                                                         *
//*                                                                     *
//*          TCPIP.ROUTING                                              *
//*          TCPIP.OPTIONS                                              *
//*          TCPIP.QUEUE                                                *
//*                                                                     *
//*  All three are assumed to have been created and initialized         *
//*  prior to the use of this JCL.                                     *
//*                                                                    *
//* -------------------------------------------------------------------*
//*  Step 1 :                                                          *
//*                                                                    *
//*    This step assembles the File Load Program                       *
//*                                                                    *
//*    This is the area where you will enter the routing and options   *
//*    information.                                                    *
//* -------------------------------------------------------------------*
//ASM       EXEC PGM=IEV90,PARM='OBJECT,TERM',REGION=1024K
//SYSLIB    DD DSN=TCPIP.SEZACMAC,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//          DD DSN=SYS1.MODGEN,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//SYSUT2    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT3    DD UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSPUNCH  DD DUMMY
//SYSLIN    DD DSNAME=&&OBJSET,DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(80,(500,50))
//SYSTERM   DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
* --------------------------------------------------------------------*
*     Must start with INITIAL parm                                    *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=INITIAL,                                         X
               PRGNAME=ADDRTOPT,                                       X
               MODE=UPDATE
* --------------------------------------------------------------------*
*     Example of a normal route for VTAM                              *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=ROUTING,                                         X
               MAJKEY=DLU10100,                                        X
               MINKEY=VTAM,                                            X
               RETAINS=10,                                             X
               RETAINU=20,                                             X
               RETRYT=5,                                               X
               RETRYL=2,                                               X
               NDEST=1,                                                X
               OPTNAME=LUOPTN3,                                        X
               LUCLASS=(1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,64),       X
               INAME=9.67.112.25,                                      X
               PNAME=LPT1
* --------------------------------------------------------------------*
*     Example of a normal route for the JES writer                    *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=ROUTING,                                         X
               MAJKEY=LOCAL,                                           X
               MINKEY=A1185,                                           X
               RETAINS=5,                                              X
               RETAINU=10,                                             X
               RETRYT=5,                                               X
               RETRYL=2,                                               X
               NDEST=1,                                                X
               OPTNAME=OPTION3,                                        X
               INAME=MVS1.TCP.RALEIGH.IBM.COM,                         X
               PNAME=LPT1
* --------------------------------------------------------------------*
*     Example of an options entry                                     *
* --------------------------------------------------------------------*
*     LUOPTN3                                                         *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=OPTIONS,                                         X
               OPTNAME=LUOPTN3,                                        X
               OPTIONS=TRACE
* --------------------------------------------------------------------*
*     OPTION3                                                         *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=OPTIONS,                                         X
               OPTNAME=OPTION3,                                        X
               OPTIONS='TRACE CLASS E'
* --------------------------------------------------------------------*
*     Must end with FINAL parm                                        *
* --------------------------------------------------------------------*
         EZAPPFL TYPE=FINAL
         END
/*
//* -------------------------------------------------------------------*
//*  Step 2 :                                                          *
//*                                                                    *
//*    This step links the File Load Program                           *
//*                                                                    *
//* -------------------------------------------------------------------*
//LKED    EXEC PGM=IEWL,PARM='LIST,MAP,XREF',
//             REGION=512K,COND=(4,LT)
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD SPACE=(CYL,(5,1)),DISP=(NEW,PASS),UNIT=SYSDA
//SYSLMOD   DD DSNAME=&&GOSET(GO),DISP=(MOD,PASS),UNIT=SYSDA,
//             SPACE=(1024,(50,1,1))
//SYSLIN    DD DSNAME=&&OBJSET,DISP=(OLD,DELETE)
//LINK.SYSIN  DD DUMMY
//* -------------------------------------------------------------------*
//*  Step 3 :                                                          *
//*                                                                    *
//*    This step executes the File Load Program                        *
//*                                                                    *
//* -------------------------------------------------------------------*
//LOAD     EXEC PGM=*.LKED.SYSLMOD,REGION=512K,COND=(4,LT)
//EZAPPRF  DD   DSN=TCPIP.ROUTING,DISP=SHR
//EZAPPOF  DD   DSN=TCPIP.OPTIONS,DISP=SHR
//EZAPPQF  DD   DSN=TCPIP.QUEUE,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSABEND DD   SYSOUT=*
//SYSMDUMP DD   SYSOUT=*
```

Chapter 6. Maintaining the Network Print Facility files with ISPF

After the routing, options, and queue files have been created and initialized, you can use the Network Print FacilityISPF interface panels to add or update the routing and options records and manage your print queues.

This section shows you how to:

- Enable the NPF interface
- Set defaults for your interactive session
- Manage the print queue by browsing, updating, and deleting queue file records
- Maintain the routing and options files by adding, copying, updating, and deleting records

Enabling the Network Print Facility ISPF interface

The Network Print Facility ISPF interface panels can be used to add or update the routing and options records and manage your print queues. You can enable the Network Print Facility ISPF interface by adding NPF to the ISPF Primary Option Menu.

Procedure

To enable the Network Print Facility ISPF interface, perform the following steps.

1. Provide library access

   You must provide access to the NPF ISPF libraries. You can do this by modifying the TSO logon procedures or running a CLIST.
2. Define default data set names

   You can define data set names for the routing, options, queue, and trace files that are accessed during ISPF sessions. You can use an initialization CLIST to set different defaults for each user, for specific groups of users, or for all users.
3. Add NPF to the ISPF Primary Option Menu

   To be able to select the NPF interface from your ISPF Primary Option Menu, you need to update the menu and processing sections of the ISR@PRIM module.

Results

Note: You must be able to scroll forward and back in the ISPF interface to be able to access specific broadcast records. Be sure that your keyboard has specific keys for **Page Up** and **Page Down** or that you have set PF keys for these functions using option **0.3** on the ISPF Primary Option Menu. UP or FORWARD will work for scrolling forward. DOWN, BACK, or BACKWARD will work for scrolling back.

Providing library access

You must provide access to the NPF ISPF libraries. You can do this by either:

- Adding DD statements to your TSO logon procedure
- Allocating the data sets with a CLIST

Network Print Facility ISPF data sets

The following libraries are partitioned data sets required for operation of the NPF ISPF interface in any MVS/TSO environment.

SEZAPENU (Panel Library)

hlq.SEZAPENU includes all the data panels required for NPF. The members in this library are:

- EZAPPN (Network Print Facility Data Panels)
- EZAPPH (Network Print Facility Help Panels)

SEZAMENU (Message Library)

hlq.SEZAMENU includes all the messages for NPF. Its member is EZYPR (Network Print Facility ISPF messages).

Using the TSO logon procedure

One method of providing access to the NPF ISPF libraries is to specify the NPF ISPF data sets in the TSO logon procedure.

Add the following DD statements to your TSO logon procedure and replace TCPIP with your installation's high-level qualifier.

```
   //ISPPLIB         DD   DSN=TCPIP.SEZAPENU,DISP=SHR
   //ISPMLIB         DD   DSN=TCPIP.SEZAMENU,DISP=SHR
```

Figure 9. TSO logon procedure

If you need further information about the TSO logon procedure, see the *ISPF Dialog Management Guide and Reference*

Using a CLIST

Another method of providing access to the NPF ISPF interface is to run a CLIST to allocate the NPF ISPF data sets. Copy *hlq*.SEZAINST(NPFINIT) into your system CLIST library and make changes to the sections that allocate and deallocate the libraries.

Defining default data set names

If you do not define your own defaults, the NPF ISPF interface will use the following data set names as the defaults:

Routing file
TCPIP.ROUTING

Options file
TCPIP.OPTIONS

Queue file
TCPIP.QUEUE

Trace file
*userid.*NPFISPF.TRACE

You can change the default data set names for these files by updating and implementing the appropriate section in the initialization CLIST. You can set one default for all users or set different defaults for individual users.

Whether you use the system settings or define your own values, the NPF interface will retrieve the defaults at the start of an interactive session and display them on the panels. These values represent the actual data sets the interface will access. The users can change to other data sets by keying over the displayed values or they can go back to the initial defaults by clearing the fields.

To change the defaults, copy *hlq*.SEZAINST(NPFINIT) into your system CLIST library and make changes by following the instructions for this section.

You can run this CLIST as a stand-alone process or invoke it from the ISPF Primary Option Menu. See "Updating ISR@PRIM processing section" on page 64 for instructions on invoking it from the menu.

```
PROC 0
CONTROL NOFLUSH NOMSG MAIN
/*********************************************************************/
/*                                                                   */
/*  CLIST NAME:      NPFINIT                                         */
/*                                                                   */
/*  COPYRIGHT = NONE.                                                */
/*                                                                   */
/*  SMP/E Distribution Name: EZAEC050                                */
/*                                                                   */
/*  DESCRIPTION:                                                     */
/*     This is a sample CLIST which does initialization processing   */
/*     for NPF's ISPF panel interface and then invokes the NPF main  */
/*     panel.  This CLIST can be modified as necessary to match      */
/*     specific installation requirements.                           */
/*                                                                   */
/*     This CLIST can be invoked via a stand-alone TSO command,      */
/*     or it can be invoked from the system's main ISPF panel.       */
/*                                                                   */
/*     Much of the code within this CLIST is optional and can be     */
/*     deleted if desired.  This optional code is divided into       */
/*     3 sections as follows:                                        */
/*                                                                   */
/*     *  Section 1:  Sets up default file names for routing,        */
/*        options, queue and trace file processing on NPF's ISPF     */
/*        panel interface.  For access to each type of file, the     */
/*        default name will be used when no previously-used name     */
/*        for that file has been saved by the current user id.       */
/*                                                                   */
/*     *  Section 2:  Allocates the libraries containing the panels  */
/*        and modules required by NPF's ISPF panel interface.        */
/*                                                                   */
/*     *  Section 3:  Restores the ISPF environment by deallocating  */
/*        the libraries allocated by the code in Section 2.          */
/*                                                                   */
/*     Set the &DODFLTS variable to 'Y' or 'N' to indicate           */
/*     whether the default-setting logic should be executed.         */
/*                                                                   */
/*     Set the &DOLIBS variable to 'Y' or 'N' to indicate            */
/*     whether the library-allocation logic should be executed.      */
/*                                                                   */
/*********************************************************************/
  SET &DODFLTS = Y                  /* Set defaults                  */
  SET &DOLIBS = N                   /* Do not allocate libraries     */
/*********************************************************************/
/*                                                                   */
/*         O P T I O N A L   C O D E   -   S E C T I O N   1         */
/*                                                                   */
/*  The code in this section initializes default file names for      */
/*  NPF's routing, options, queue and/or trace files.  These default */
/*  names can be set the same for all users, or they can be set to   */
/*  different values for different user ids.                         */
/*                                                                   */
/*  If this section is deleted or bypassed, NPF's panel code         */
/*  will set its own hard-coded default values as documented in      */
/*  the NPF manual.                                                  */
/*                                                                   */
/*  Notes:                                                           */
/*                                                                   */
/*  1. The &CLSTUPDT# variable below controls whether this CLIST's   */
/*     default-setting logic is executed or bypassed when the CLIST  */
/*     gets control under a particular user id:                      */
/*                                                                   */
/*     *  Set &CLSTUPDT# to a new and higher value each time this    */
/*        CLIST's default-setting logic is changed.                  */
/*                                                                   */
/*     *  As each user id executes the CLIST, the &CLSTUPDT# value   */
/*        will be compared to the value saved when that user id's    */
/*        defaults were last updated:                                */
/*                                                                   */
/*        -  An updated &CLSTUPDT# value will cause the CLIST to     */
/*           execute its default-setting logic.                      */
/*                                                                   */
/*        -  An unchanged &CLSTUPDT# value will cause the CLIST      */
/*           to bypass ist default-setting logic.                    */
/*                                                                   */
/*  2. This sample CLIST assumes that you want default file          */
/*     names as follows:                                             */
/*                                                                   */
/*     *  For the routing file:                                      */
/*        - For user id = USER6, file name = XYZ.PERS.ROUTING        */
/*        - For all other users, file name = TCPIP.NPF.ROUTING       */
/*                                                                   */
/*     *  For the options file:                                      */
/*        - For user id = USER6, file name = XYZ.PERS.OPTIONS        */
/*        - For all other users, file name = TCPIP.NPF.OPTIONS       */
/*                                                                   */
/*     *  For the queue file:                                        */
/*        - For user id = USER2 or USER3 or USER17,                  */
/*                               file name = ABC.ACCT.QUEUE          */
/*        - For user id = USER5, file name = PQR.PAYROLL.QUEUE       */
/*        - For user id = USER6, file name = XYZ.PERS.QUEUE          */
/*        - For all other users, file name = TCPIP.NPF.QUEUE         */
/*                                                                   */
/*     *  For the trace file:                                        */
/*        - For all users,       file name = userid.NPF.TRACE        */
/*                                                                   */
/*********************************************************************/
IF &DODFLTS = Y THEN +
  DO
    SET &CLSTUPDT# = 1                /* Set current update number   */
    ISPEXEC VGET NPFUPDT# PROFILE     /* Get last-saved update number*/
    IF &NPFUPDT# ^= &CLSTUPDT# THEN +
      DO
        SET &NPFUPDT# = &CLSTUPDT#    /* Save the current            */
        ISPEXEC VPUT NPFUPDT# PROFILE /* ... update number value     */
        /*-----------------------------------------------------------*/
        /*   Set defaults applicable to most or all user ids         */
        /*-----------------------------------------------------------*/
        SET &NPFDFLTR = TCPIP.NPF.ROUTING
        SET &NPFDFLTO = TCPIP.NPF.OPTIONS
        SET &NPFDFLTT = &SYSUID..NPF.TRACE
        /*---------------------------------------------------------*/
        /*   Set defaults for single user ids or sets of user ids  */
        /*---------------------------------------------------------*/
        SELECT (&SYSUID)
          WHEN (USER2 | USER3 | USER17) +
            SET &NPFDFLTQ = ABC.ACCT.QUEUE
          WHEN (USER5) +
            SET &NPFDFLTQ = PQR.PAYROLL.QUEUE
          WHEN (USER6) +
            DO
              SET &NPFDFLTR = XYZ.PERS.ROUTING
              SET &NPFDFLTO = XYZ.PERS.OPTIONS
              SET &NPFDFLTQ = XYZ.PERS.QUEUE
            END
          OTHERWISE +
            SET &NPFDFLTQ = TCPIP.NPF.QUEUE
        END                           /* End of SELECT               */
        /*-----------------------------------------------------------*/
        /*  Save the default values in the user id's Profile pool    */
        /*-----------------------------------------------------------*/
        ISPEXEC VPUT (NPFDFLTR NPFDFLTO NPFDFLTQ NPFDFLTT) PROFILE
      END
  END
/*********************************************************************/
/*                 E N D   O F   S E C T I O N   1                   */
/*********************************************************************/
/*********************************************************************/
/*                                                                   */
/*         O P T I O N A L   C O D E   -   S E C T I O N   2         */
/*                                                                   */
/*  The code in this section allocates the libraries required by     */
/*  NPF's ISPF panel interface.                                      */
/*                                                                   */
/*  Note:  This library-allocation function can also be accomplished */
/*  by adding appropriate DD cards to each user id's TSO logon       */
/*  procedure.  If that technique is used, the code in Sections 2    */
/*  and 3 should be bypassed or deleted from this CLIST.             */
/*                                                                   */
/*********************************************************************/
IF &DOLIBS = Y THEN +
  DO
    /*---------------------------------------------------------------*/
    /*   NPF libraries concatentation                                */
    /*---------------------------------------------------------------*/
    ISPEXEC LIBDEF ISPPLIB DATASET ID('TCPIP.SEZAPENU') COND
    ISPEXEC LIBDEF ISPMLIB DATASET ID('TCPIP.SEZAMENU') COND
  END
/*********************************************************************/
/*                 E N D   O F   S E C T I O N   2                   */
/*********************************************************************/
/*-------------------------------------------------------------------*/
/*                    R E Q U I R E D   C O D E                      */
/*-------------------------------------------------------------------*/
ISPEXEC SELECT PANEL(EZAPPMP)      /* Invoke the NPF main panel      */
/*********************************************************************/
/*                                                                   */
/*         O P T I O N A L   C O D E   -   S E C T I O N   3         */
/*                                                                   */
/*  The code in this section restores the ISPF library environment   */
/*                                                                   */
/*  Note:  This section is required if Section 2 is present, but     */
/*  should be bypassed or deleted if Section 2 is bypassed or deleted*/
/*                                                                   */
/*********************************************************************/
IF &DOLIBS = Y THEN +
  DO
    /*---------------------------------------------------------------*/
    /*   Restore the ISPF environment                                */
    /*---------------------------------------------------------------*/
    ISPEXEC LIBDEF ISPPLIB
    ISPEXEC LIBDEF ISPMLIB
  END
/*********************************************************************/
/*                 E N D   O F   S E C T I O N   3                   */
/*********************************************************************/
/*-------------------------------------------------------------------*/
/*                    R E Q U I R E D   C O D E                      */
/*-------------------------------------------------------------------*/
RETURN                             /* Return to invoker              */
```

Adding NPF to the ISPF primary option menu

ISR@PRIM is a member in the ISPPLIB library. If you want your users to have access to the NPF ISPF interface from the ISPF Primary Option Menu, you must update ISR@PRIM in two places:

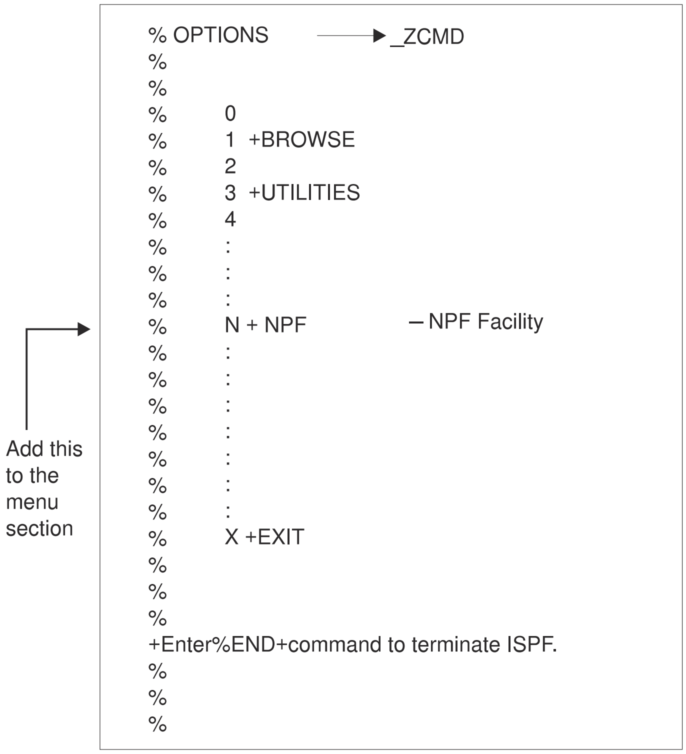
- In the menu section (Part 1 of ISR@PRIM) to have an option for NPF to appear on the ISPF Primary Option Menu. See the example shown in Figure 10 on page 64.
- In the processing section (Part 2 of ISR@PRIM) to have the selection invoke the NPF interface. You can optionally have the selection run the initialization CLIST before invoking the NPF interface. See the examples shown in Figure 11 on page 65 and Figure 12 on page 66.

After you have updated ISR@PRIM, the option you added for NPF will appear on the ISPF Primary Option Menu following the next ISPF logon.

Updating ISR@PRIM menu section

To have an option for the NPF ISPF interface appear on the ISPF Primary Option Menu, add an entry in the menu section of ISR@PRIM.

You can choose any value. This example uses the letter **N**.



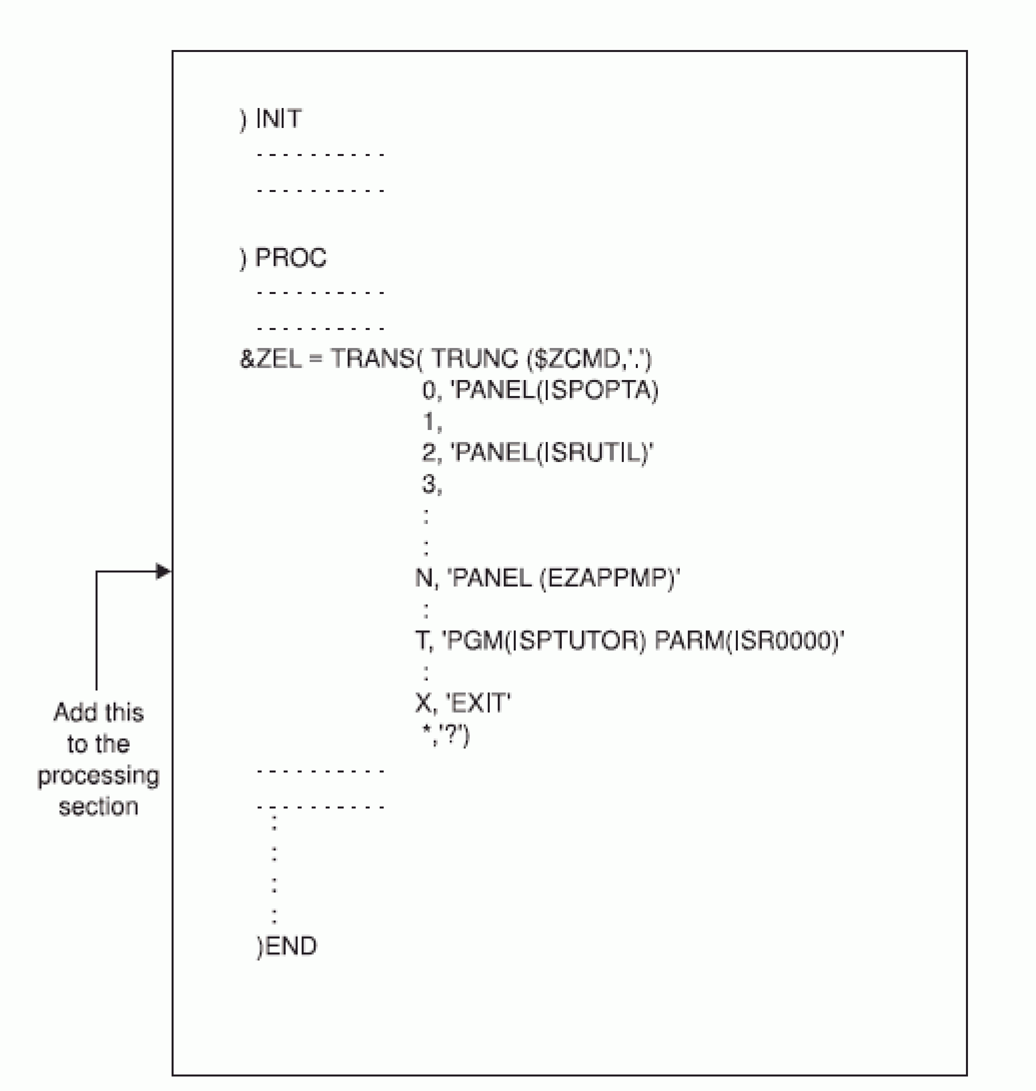
```
% OPTIONS          _ZCMD
%
%
%       0
%       1  +BROWSE
%       2
%       3  +UTILITIES
%       4
%       :
%       :
%       :
%       N + NPF             – NPF Facility
%       :
%       :
%       :
%       :
%       :
%       :
%       :
%       X +EXIT
%
%
%
+Enter%END+command to terminate ISPF.
%
%
%
```

Add this to the menu section

Figure 10. ISPF primary option menu ISR@PRIM (Part 1 - menu section)

Updating ISR@PRIM processing section

If you ***do not*** want to run the initialization CLIST before invoking the NPF interface, use the following example to update the processing section of ISR@PRIM. In this example, the selection **N** invokes the NPF main menu panel, EZAPPMP.

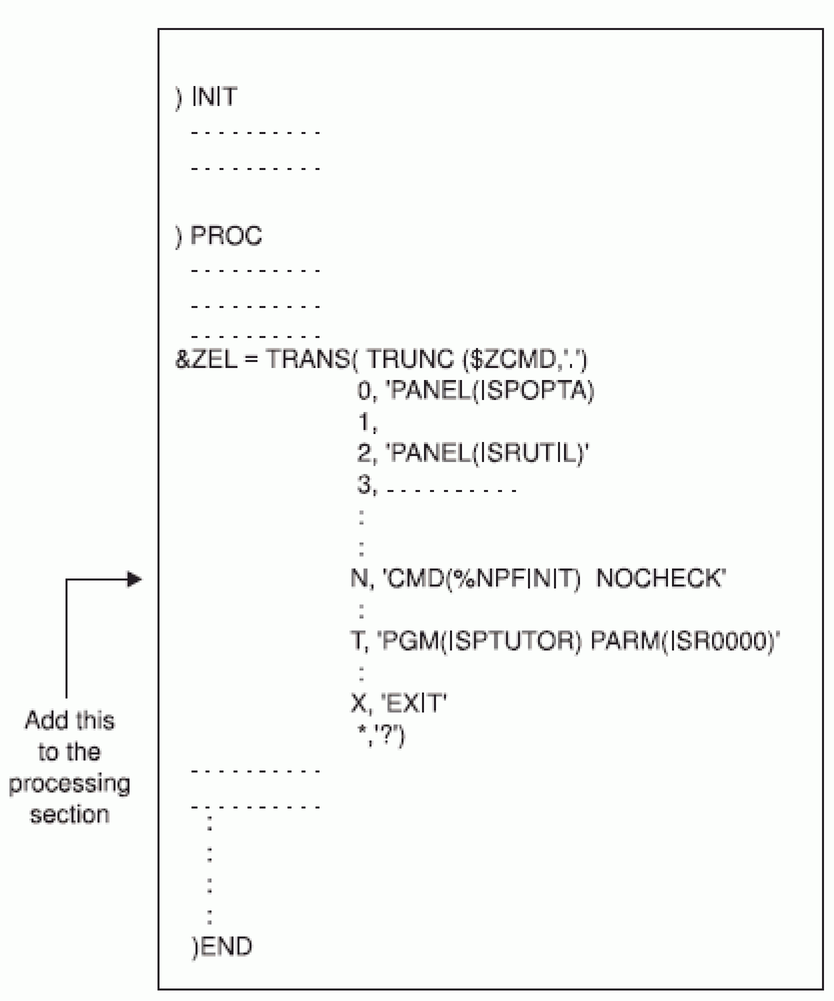


```
) INIT
  - - - - - - - - - -
  - - - - - - - - - -

) PROC
  - - - - - - - - - -
  - - - - - - - - - -
&ZEL = TRANS( TRUNC ($ZCMD,'.')
               0, 'PANEL(ISPOPTA)
               1,
               2, 'PANEL(ISRUTIL)'
               3,
               :
               :
              N, 'PANEL (EZAPPMP)'
               :
              T, 'PGM(ISPTUTOR) PARM(ISR0000)'
               :
              X, 'EXIT'
               *,'?')
  - - - - - - - - - -
  - - - - - - - - - -
   :
   :
   :
   :
  )END
```

Add this to the processing section

Figure 11. ISPF primary option menu ISR@PRIM (Part 2a.- processing section)

If you ***do*** want to run an initialization CLIST before invoking the NPF interface, use the following example to update the processing section of ISR@PRIM. In this example, the selection **N** invokes the NPFINIT CLIST, which in turn invokes the NPF main menu panel, EZAPPMP.



```
) INIT
  - - - - - - - - - -
  - - - - - - - - - -

) PROC
  - - - - - - - - - -
  - - - - - - - - - -
  - - - - - - - - - -
&ZEL = TRANS( TRUNC ($ZCMD,'.')
                    0, 'PANEL(ISPOPTA)
                    1,
                    2, 'PANEL(ISRUTIL)'
                    3, - - - - - - - - - -
                    :
                    :
                   N, 'CMD(%NPFINIT)  NOCHECK'
                    :
                   T, 'PGM(ISPTUTOR) PARM(ISR0000)'
                    :
                   X, 'EXIT'
                    *,'?')
  - - - - - - - - - -
  - - - - - - - - - -
   :
   :
   :
   :
  )END
```

Add this to the processing section

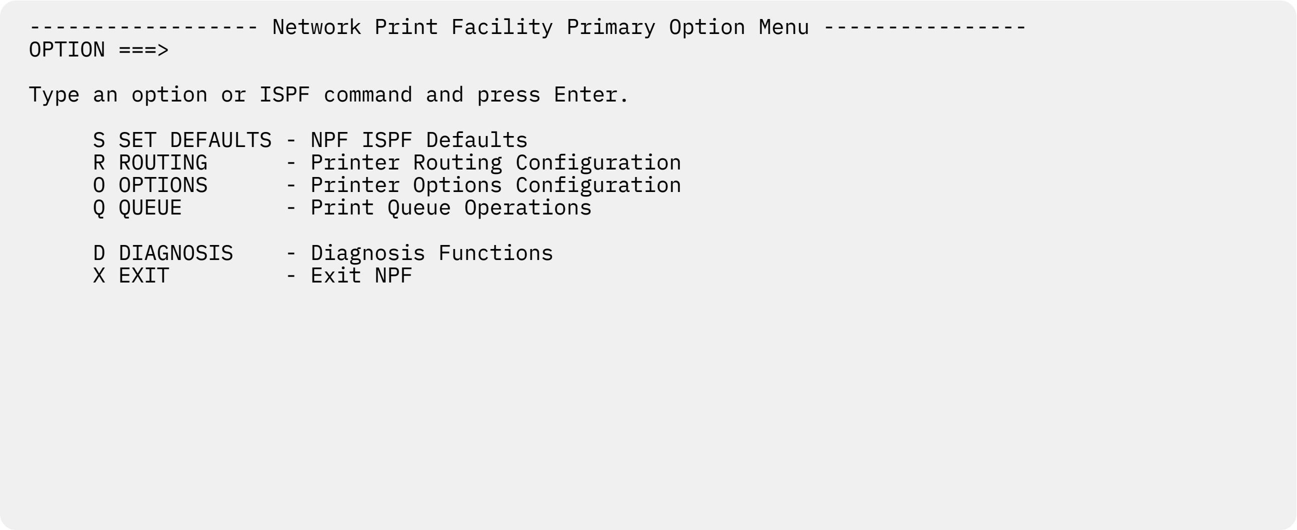
Figure 12. ISPF primary option menu ISR@PRIM (Part 2b.- processing section)

Starting an NPF ISPF interface session

The NPF ISPF interface allows you to set options for your interactive session, maintain the routing and options files, and manage the print queue.

- You can configure the interface by setting the delete confirmation option for the routing, options, and queue files. You can also turn NPF tracing on or off.
- You can maintain the routing and options files by adding, copying, updating, or deleting records.
- You can manage your print queues by browsing the queue file, updating records to affect when (and if) the system will try to resend a print job, rerouting print jobs, and requesting queue records to be deleted.

Begin all NPF interactive sessions at the Network Print Facility Primary Option Menu shown in Figure 13 on page 67. To display this menu, select your installation-specific NPF option from the ISPF Primary Option Menu. See your system administrator if this option is not on your ISPF menu.



```
------------------ Network Print Facility Primary Option Menu ----------------
OPTION ===>

Type an option or ISPF command and press Enter.

     S SET DEFAULTS - NPF ISPF Defaults
     R ROUTING      - Printer Routing Configuration
     O OPTIONS      - Printer Options Configuration
     Q QUEUE        - Print Queue Operations

     D DIAGNOSIS    - Diagnosis Functions
     X EXIT         - Exit NPF
```

Figure 13. Network Print Facility Primary Option Menu (EZAPPMP)

From this menu you can proceed to all the NPF panels and perform the tasks required to maintain your files and print queues.

Task	**Selection**
Turn delete confirmation on or off	`S SET DEFAULTS`
Turn NPF ISPF tracing on or off	`D DIAGNOSIS`
Add, update, copy or delete printer routes	`R ROUTING`
Add, update, copy or delete printer options	`O OPTIONS`
Maintain your print queues	`Q QUEUE`
Return to the ISPF Primary Option Menu	`X EXIT`

Helpful reminders

As you work with the NPF ISPF interface, keep these ideas in mind:

- Before you use the panel interface, ensure you have created the options, routing, and queue files correctly and that they contain valid information.
- Only people with system administrator or network administrator authority should change the routing file or options file records.
- Use only the NPF ISPF interface, not the ISPF VSAM editing option, to edit the options, routing, and queue files.
- Keep a list of your general use printers for reference.
- Plan to carefully record the keys (MAJOR, MINOR and OPTIONS NAME) of the records you create, as NPF does not provide a list of records by key field.
- HOST NAME/IP ADDR (Host Name or IP Address) and PRINTER NAME are mixed case fields. All other fields are uppercase.
- There is online help available for quick assistance on each NPF panel, on each input field, and on each message.
- The characters you see in the fields on the panels in this book have these meanings:

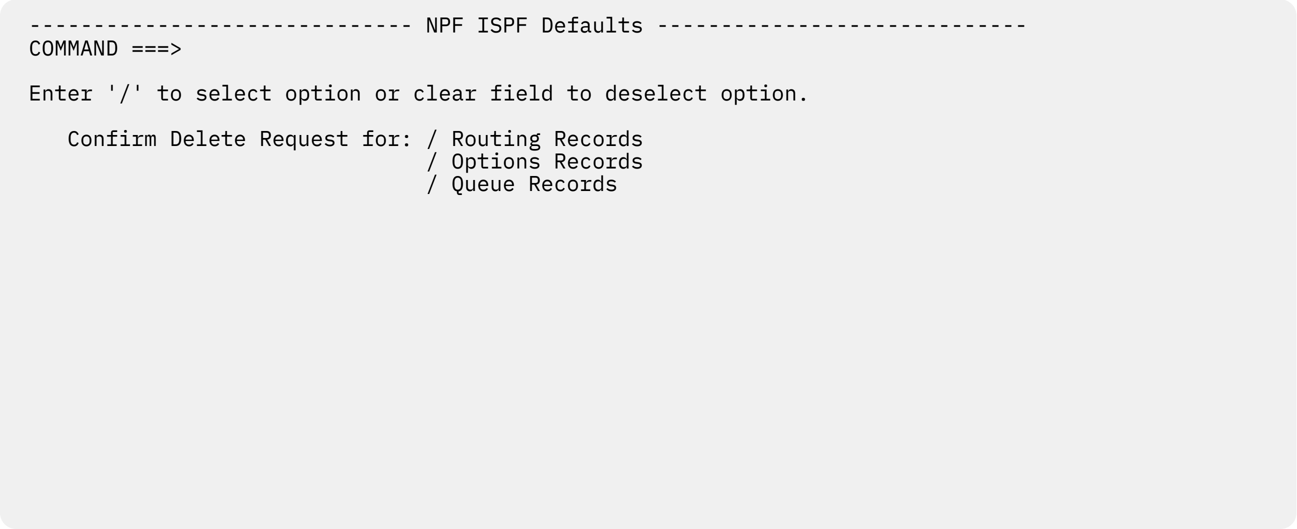
  XXXXXX
  The field is filled in by the system. You cannot change the values.

  SSSSSS
  The field displays a default or current value. You can change these values if you choose.

(blank space)
Some of these fields are required and you must enter data; others are optional and can remain blank.

Setting the NPF ISPF defaults

To display the NPF ISPF Defaults panel shown in Figure 14 on page 68, enter **S** on the Network Print Facility Primary Option menu.



```
------------------------------ NPF ISPF Defaults ------------------------------
COMMAND ===>

Enter '/' to select option or clear field to deselect option.

   Confirm Delete Request for: / Routing Records
                               / Options Records
                               / Queue Records
```

Figure 14. NPF ISPF Defaults (EZAPPDCP)

This panel allows you to choose whether or not you want to be prompted with a confirmation window when you delete records from your files. You can set these options at any time during your interactive session.

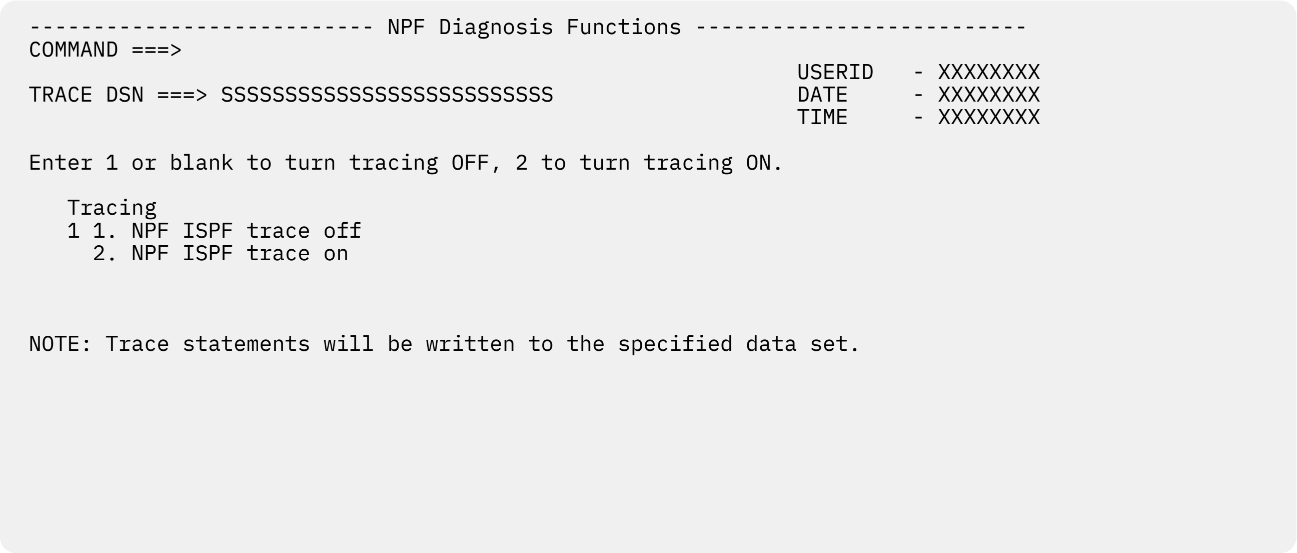
You have the option of requesting delete confirmation panels for each of three NPF files. Although delete confirmation panels provide an extra step in the deletion process and take more time, they allow you to change your mind or back out of a request made in error.

To enable this option, enter a **/** (backslash) next to your choices. Every time you delete these types of records, you will get a panel asking you to verify your request. The delete confirmation panel will ***not*** appear when the queue manager deletes queue file records that have expired.

To stop delete confirmation, clear the selection field.

Setting trace options

To display the NPF Diagnosis Functions panel shown in Figure 15 on page 69, enter **D** on the Network Print Facility Primary Option menu.



```
-------------------------- NPF Diagnosis Functions -------------------------
COMMAND ===>
                                                            USERID   - XXXXXXXX
TRACE DSN ===> SSSSSSSSSSSSSSSSSSSSSSSSSS                   DATE     - XXXXXXXX
                                                            TIME     - XXXXXXXX

Enter 1 or blank to turn tracing OFF, 2 to turn tracing ON.

   Tracing
   1 1. NPF ISPF trace off
     2. NPF ISPF trace on


NOTE: Trace statements will be written to the specified data set.
```

Figure 15. NPF Diagnosis Functions (EZAPPDGP)

This panel allows you to turn the NPF trace facility on or off. This function traces only activity related to NPF internal processing and writes the trace records to the data set displayed on the panel.

The name of this data set is either the default trace data set or the last data set you entered in this field. The default can be changed by a system programmer in an initialization CLIST. See "Defining default data set names" on page 60 for details. You can key over this name to use another trace data set.

Typically, you would turn this trace on for short periods of time while diagnosing problems and collecting information to forward to the IBM Software Support Center.

If the data set specified on this panel does exist, the system will append the trace records to it. If it does not exist, the system will create it ONLY if the high-level qualifier is the same as the user's TSO user ID. Otherwise, you must allocate it as a sequential file according to the specifications given on "The trace file" on page 41. For more details on using the Network Print Facility's ISPF trace facility, see "Using the NPF ISPF trace facility" on page 126.

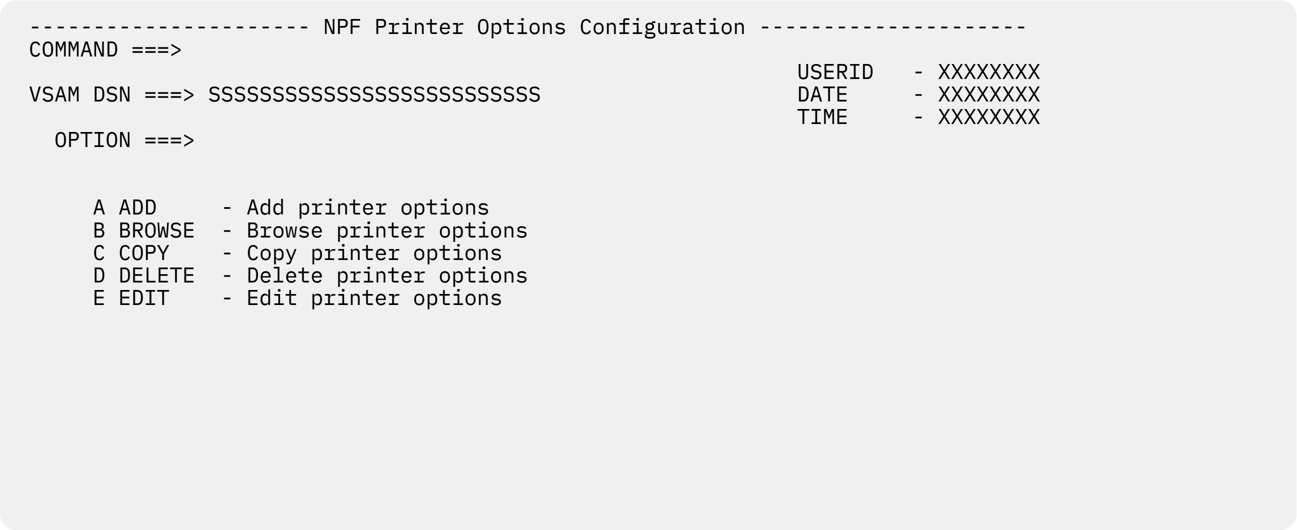
To turn tracing on, enter a **1**. To turn tracing off, enter a **2** or leave the field blank.

To write to another data set, enter the new name in this field. To go back to the default, clear the field and press **Enter**.

Maintaining the options file

The NPF ISPF interface allows you to maintain your options file by providing you with panels to add, browse, copy, delete, or edit options file records.

To display the NPF Printer Options Configuration panel shown in Figure 16 on page 70, enter **O** on the Network Print Facility Primary Option menu.



```
---------------------- NPF Printer Options Configuration ---------------------
COMMAND ===>
                                                            USERID   - XXXXXXXX
VSAM DSN ===> SSSSSSSSSSSSSSSSSSSSSSSSSS                    DATE     - XXXXXXXX
                                                            TIME     - XXXXXXXX
  OPTION ===>


     A ADD     - Add printer options
     B BROWSE  - Browse printer options
     C COPY    - Copy printer options
     D DELETE  - Delete printer options
     E EDIT    - Edit printer options
```

Figure 16. NPF Printer Options Configuration (EZAPPOCP)

This panel displays your user ID, and the current day and date in the format for which your system was configured. It also displays the options file (VSAM DSN) you will access.

The file displayed is either the default file or the last file you used. If you want to work with a different options file, enter a fully-qualified data set name in this field. If you want to reset this field back to the default, clear the field and press **ENTER**.

From this panel you can proceed to add, browse, copy, delete, or edit records in the specified options file.

Table 7. The NPF option file functions

Task	**Selection**
Add a new record to the options file	A Add printer options
Browse a record in the options file	B Browse printer options
Copy an old options file record to a new one	C Copy printer options
Delete an existing record from the options file	D Delete printer options
Update an existing options file record	E Edit printer options

In each of these tasks, you will be entering the OPTIONS NAME of a new or existing record and then adding or updating data on a subsequent panel.

Table 8 on page 70 shows the data you need for these panels. See this table as you work with the options file records.

Table 8. Option file ISPF input fields

ISPF Field	**Definition**
OPTIONS NAME	An alphanumeric field containing the name of the set of options. This is the key field for this options record. It must be unique and can have up to 16 alphanumeric characters. (For Hewlett Packard printers, the printer name must be text and all lowercase characters).
INPUT RECORD EXIT	The name of the input record exit, if one is used. This can be up to 8-characters. This field can be blank if no exit is used.

Table 8. Option file ISPF input fields (continued)

ISPF Field	Definition
LPR OPTIONS	A list of the LPR options with each option separated by blanks. This field can contain up to 255 characters. If the field is left blank, the default LPR options will be used for routings that specify this record. The valid LPR options are described in z/OS Communications Server: IP User's Guide and Commands.

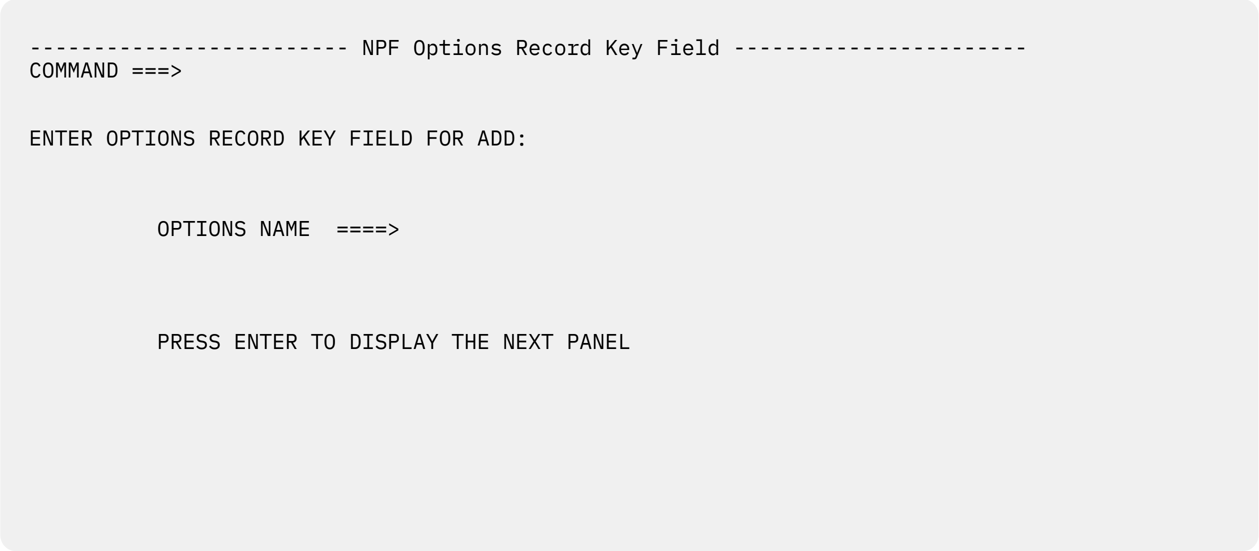
Adding records to the options file

You can add options records by creating a new record (new key field and new data).

Procedure

Follow these steps to add a new options record, creating a new record:

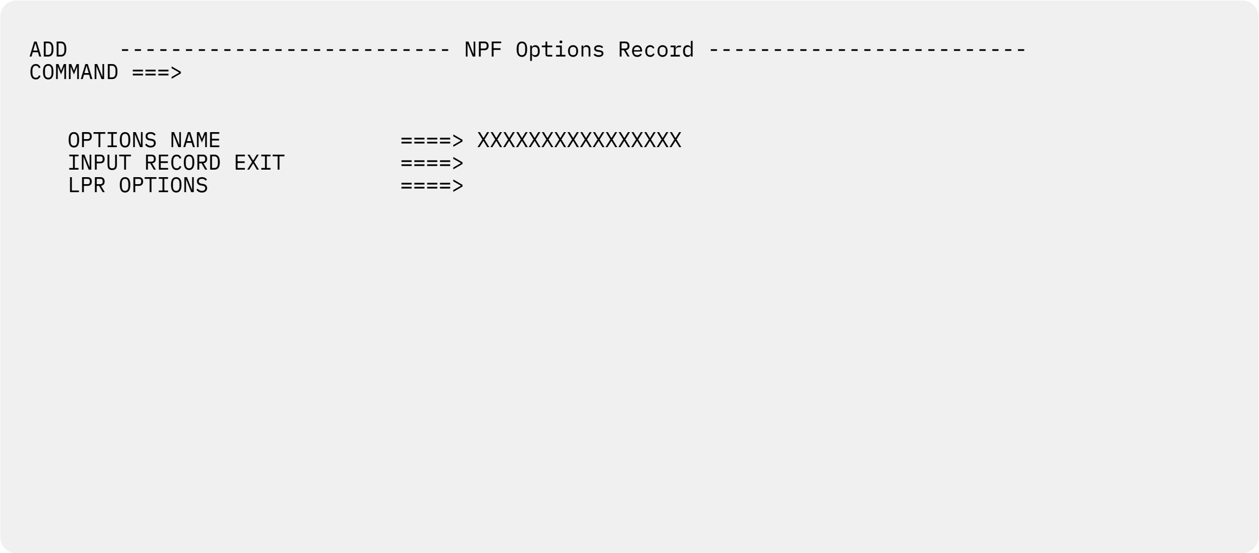
1. Select **A** from the NPF Printer Options Configuration panel to display the NPF Options Record Key Field panel (Figure 17 on page 71)



```
------------------------- NPF Options Record Key Field -----------------------
COMMAND ===>


ENTER OPTIONS RECORD KEY FIELD FOR ADD:



          OPTIONS NAME  ====>




          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 17. NPF Options Record Key Field Panel (EZAPPN1)

2. Specify the OPTIONS NAME. Enter the name you want to give the set of print options you are adding. This name will be the key field for this options record. This name must be unique and can have up to 16 alphanumeric characters. (For Hewlett Packard printers, the printer name must be text and all lowercase characters.)
3. Press **Enter** to display the NPF Options Record panel (Figure 18 on page 72).



```
ADD    ------------------------- NPF Options Record ------------------------
COMMAND ===>


   OPTIONS NAME              ====> XXXXXXXXXXXXXXXX
   INPUT RECORD EXIT         ====>
   LPR OPTIONS               ====>
```

Figure 18. NPF Options Record Panel (EZAPPN7)

4. Enter the LPR OPTIONS and an optional INPUT RECORD EXIT.

   Note: Because the OPTIONS NAME is the record key, it cannot be changed. If entered it incorrectly or misspelled it, you must cancel this record and start over.
5. To save the record, press **Enter**, and then press the END PF key or enter **END** on the command line to return.
6. To return without saving the record, enter **CANcel** on the command line.

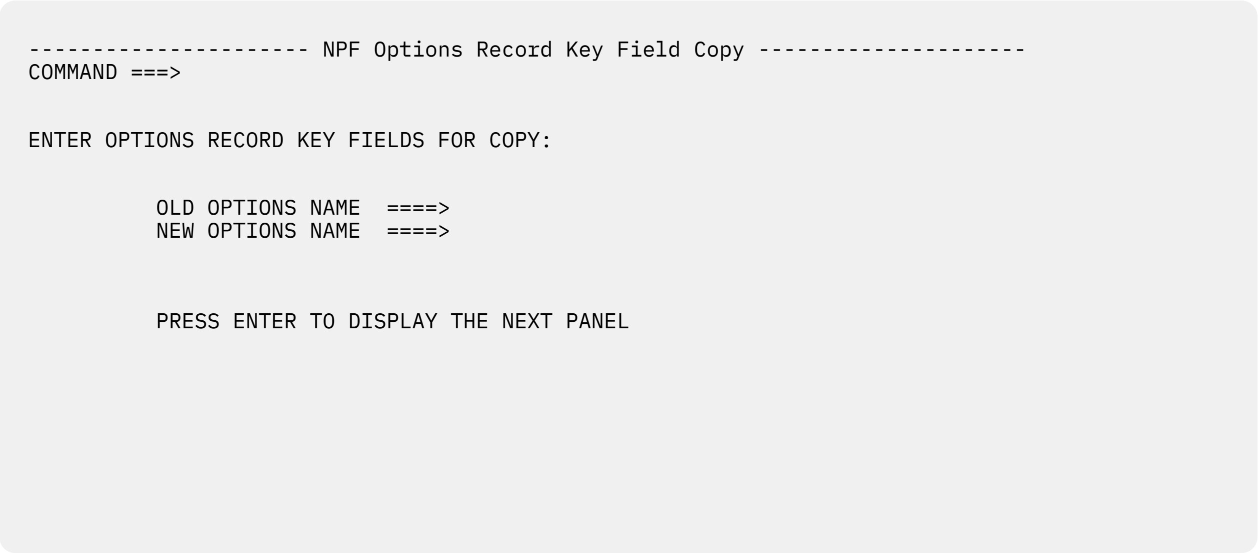
Copying records in the options file

You can create a new options file record by copying data from an existing record to a new record. You do not have to rekey long lists of LPR options.

Procedure

Follow these steps to copy an options file record from an existing record:

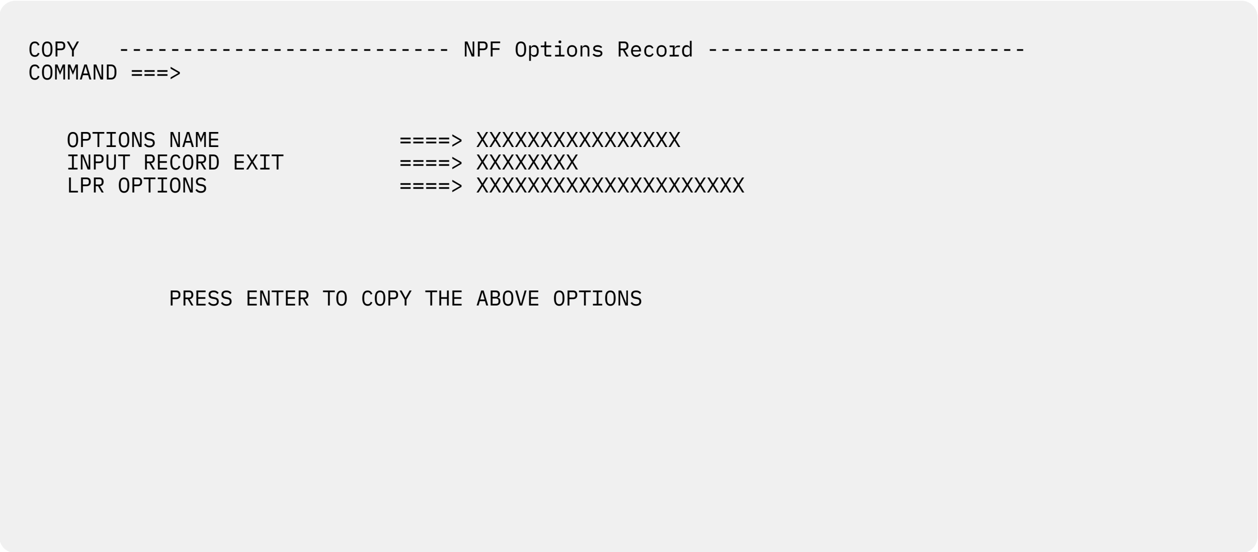
1. Select **C** from the NPF Printer Options Configuration panel to display the NPF Options Record Key Field Copy panel (Figure 19 on page 73)



```
---------------------- NPF Options Record Key Field Copy ---------------------
COMMAND ===>


ENTER OPTIONS RECORD KEY FIELDS FOR COPY:


          OLD OPTIONS NAME  ====>
          NEW OPTIONS NAME  ====>



          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 19. NPF options record key field copy panel (EZAPPN4)

2. Specify the OLD and NEW OPTIONS NAME. Enter the name of the existing record you are copying in the OLD field. Enter the name of the record you are creating in the NEW field. This is the key field for the new options record.
3. Press **Enter** to display the NPF Options Record panel (Figure 20 on page 73)



```
COPY   -------------------------- NPF Options Record -------------------------
COMMAND ===>


   OPTIONS NAME              ====> XXXXXXXXXXXXXXXX
   INPUT RECORD EXIT         ====> XXXXXXXX
   LPR OPTIONS               ====> XXXXXXXXXXXXXXXXXXXXX



           PRESS ENTER TO COPY THE ABOVE OPTIONS
```

Figure 20. NPF options record panel (EZAPPN7)

This panel displays the old key field in OPTIONS NAME and the data to be copied from the existing record. ***The information on this panel cannot be changed.***

4. Press **Enter** to copy the new options file record.

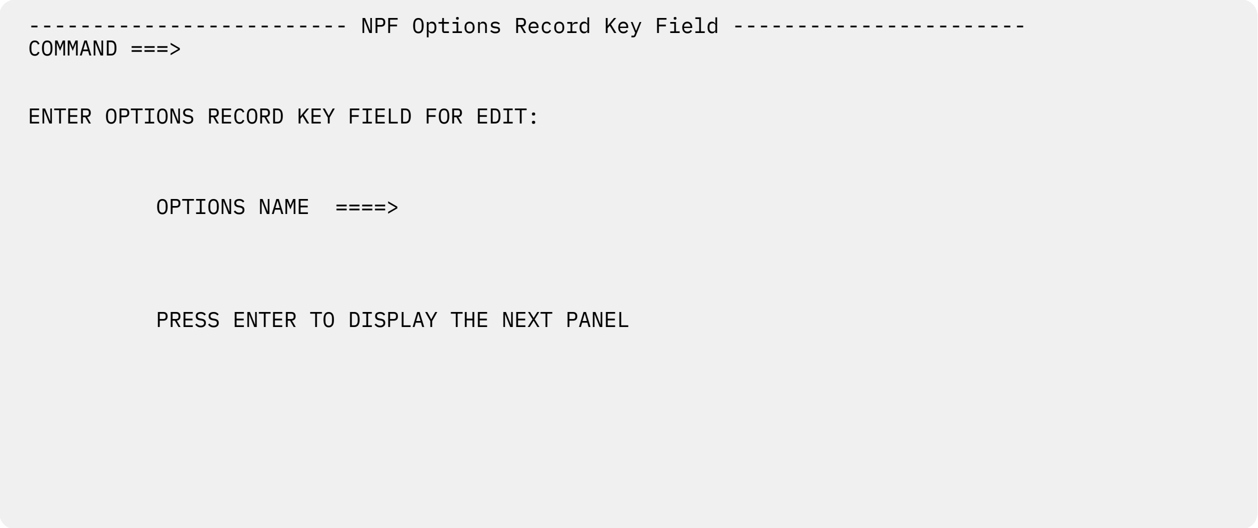
Updating records in the options file

You can update any existing options file record. Use the same rules for updating data as for entering new data in these fields.

Procedure

Follow these steps to update an options file record:

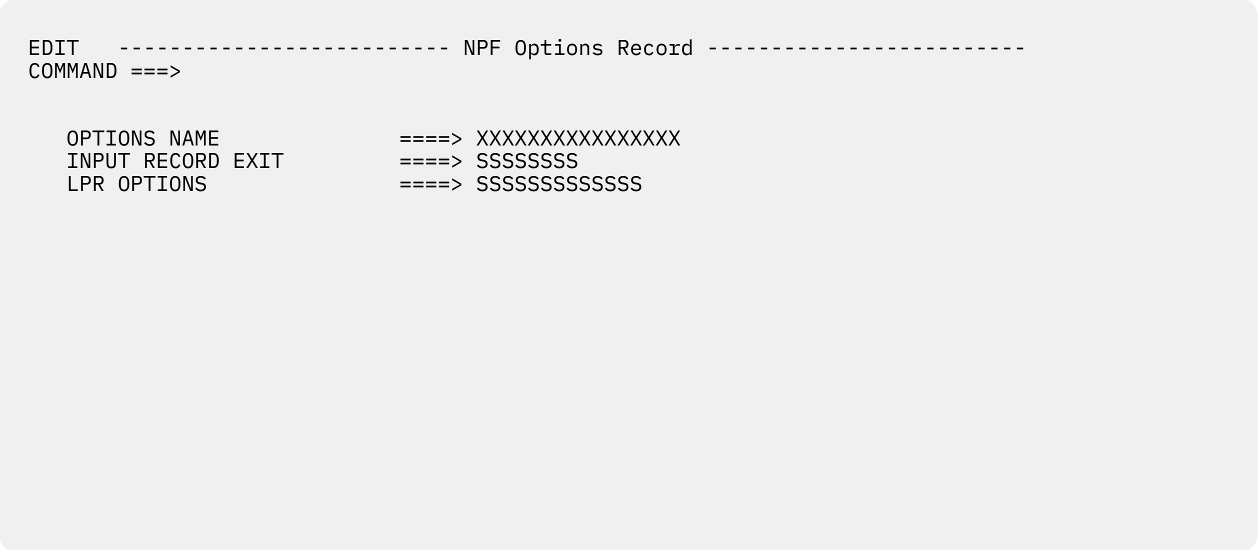
1. Select **E** from the NPF Printer Options Configuration panel to display the NPF Options Record Key Field Edit panel (Figure 21 on page 74)



```
------------------------- NPF Options Record Key Field -----------------------
COMMAND ===>


ENTER OPTIONS RECORD KEY FIELD FOR EDIT:



          OPTIONS NAME  ====>




          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 21. NPF options record key field edit panel (EZAPPN1)

2. Specify the OPTIONS NAME. Enter the name of options record you want to update. This name is the key field of this options record.
3. Press **Enter** to display the NPF Options Record Edit panel (Figure 22 on page 74)



```
EDIT   --------------------------- NPF Options Record -------------------------
COMMAND ===>


   OPTIONS NAME              ====> XXXXXXXXXXXXXXXX
   INPUT RECORD EXIT         ====> SSSSSSSS
   LPR OPTIONS               ====> SSSSSSSSSSSSS
```

Figure 22. NPF options record edit panel (EZAPPN7)

4. Change the INPUT RECORD EXIT and LPR OPTIONS as appropriate to update the record.

   Note: Because the OPTIONS NAME is the record key, it cannot be changed. If you need to correct it, you must delete this record and start over.

5. To save the changes, press **Enter**, and then press the **END** PF key or enter **END** on the command line to return.
6. To return without saving the changes, enter **CANcel** on the command line.

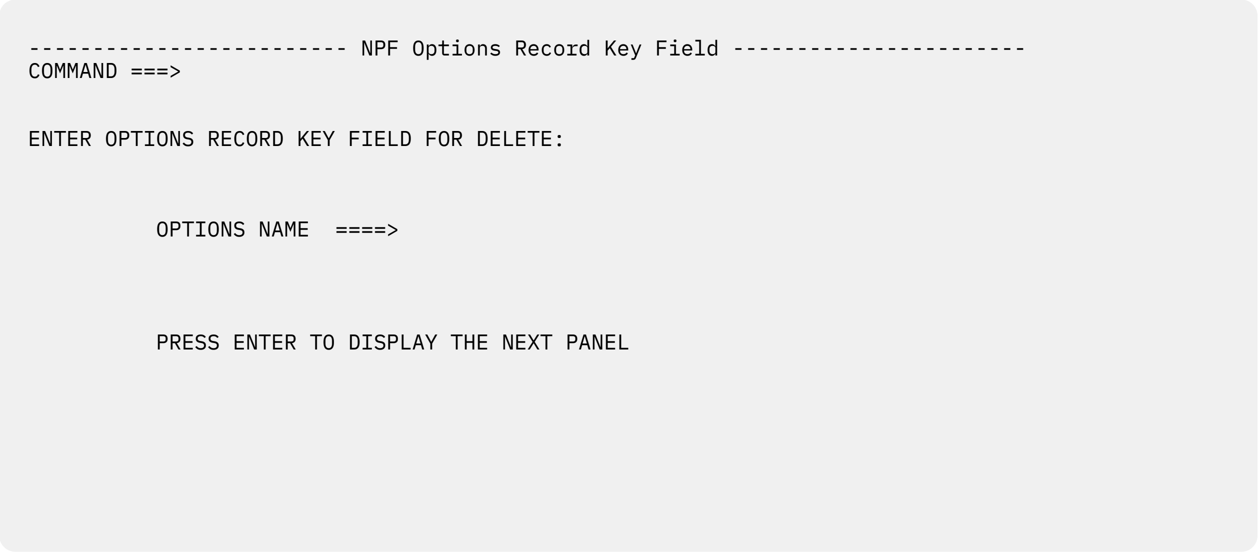
Deleting records from the options file

You should not delete any options file records that are being used by active routings. Before you delete an options file record, ensure that no routing file records have this options record specified in their OPTIONS NAME field.

Procedure

Follow these steps to delete an options file record:

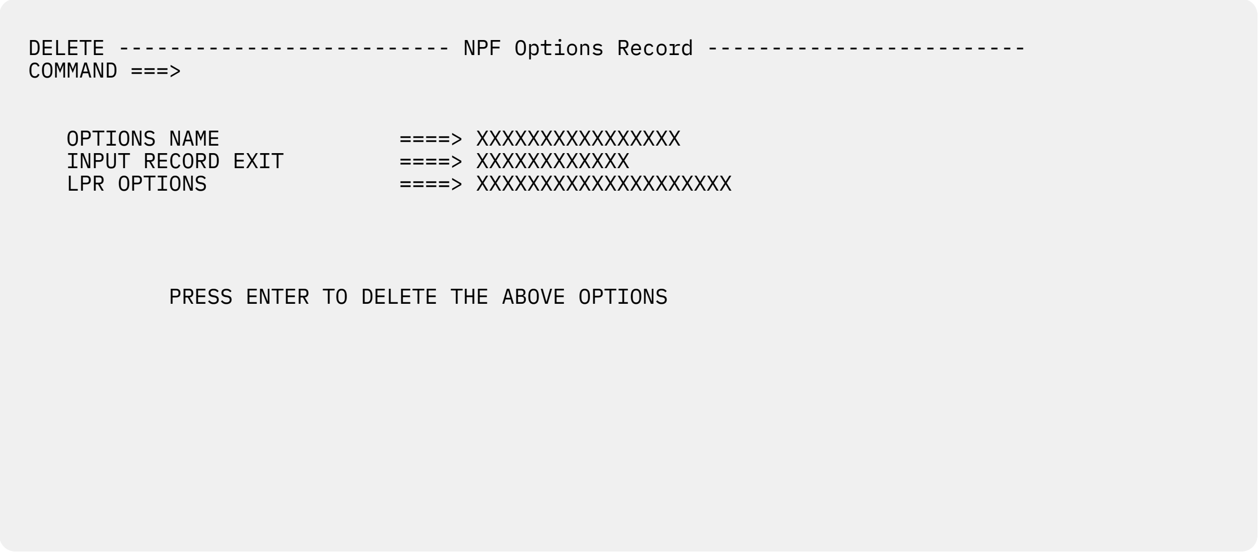
1. Select **D** from the NPF Printer Options Configuration panel to display the NPF Options Record Key Field panel (Figure 23 on page 75)



```
------------------------- NPF Options Record Key Field -----------------------
COMMAND ===>


ENTER OPTIONS RECORD KEY FIELD FOR DELETE:



          OPTIONS NAME  ====>




          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 23. NPF options record key field panel (EZAPPN1)

2. Specify the OPTIONS NAME. Enter the name of the options record you want to delete. This name is the key field for the options record.
3. Press **Enter** to display the NPF Options Record Panel (Figure 24 on page 75).



```
DELETE --------------------------- NPF Options Record -------------------------
COMMAND ===>


   OPTIONS NAME              ====> XXXXXXXXXXXXXXXX
   INPUT RECORD EXIT         ====> XXXXXXXXXXXX
   LPR OPTIONS               ====> XXXXXXXXXXXXXXXXXXXX




           PRESS ENTER TO DELETE THE ABOVE OPTIONS
```

Figure 24. NPF options record panel (EZAPPN7)

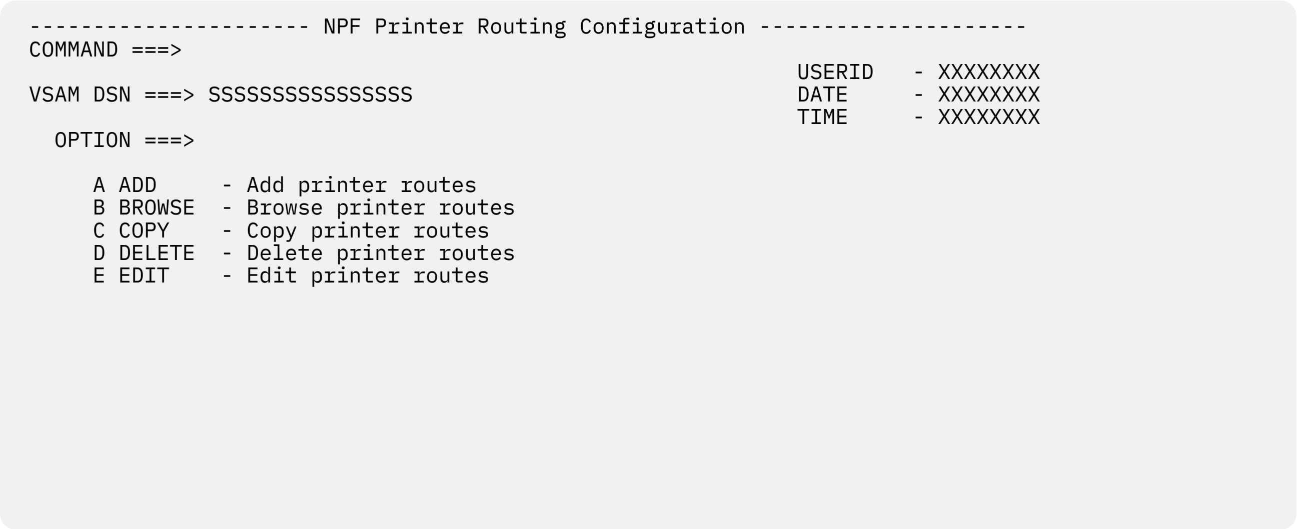
4. Verify that this is the record you want to delete.
5. Press **Enter** to delete it.

   If you requested delete confirmation for the options file, you will get a panel that allows you to continue or stop the deletion. See "Setting the NPF ISPF defaults" on page 68 for instructions to turn delete confirmation on and off.

Maintaining the routing file

The NPF ISPF interface allows you to maintain your routing file by providing you with panels to browse, add, copy, update, or delete routing file records.

To display the NPF Printer Routing Configuration panel shown in Figure 25 on page 76, enter **R** on the Network Print Facility Primary Option menu.



```
---------------------- NPF Printer Routing Configuration ---------------------
COMMAND ===>
                                                            USERID   - XXXXXXXX
VSAM DSN ===> SSSSSSSSSSSSSSSS                              DATE     - XXXXXXXX
                                                            TIME     - XXXXXXXX
  OPTION ===>

     A ADD      - Add printer routes
     B BROWSE   - Browse printer routes
     C COPY     - Copy printer routes
     D DELETE   - Delete printer routes
     E EDIT     - Edit printer routes
```

Figure 25. NPF printer routing configuration panel (EZAPPRCP)

This panel displays your user ID, and the current day and date in the format for which your system was configured. It also displays the routing file (VSAM DSN) you will access.

The file displayed is either the default file or the last file you used. If you want to work with a different routing file, enter a fully-qualified data set name in this field. If you want to reset this field back to the default, clear the field and press **ENTER**.

From this panel you can proceed to browse, add, update, or delete records in the specified routing file.

Note: This file must be created and initialized before you can perform any of these tasks.

Table 9. The NPF routing file functions

Task	Selection
Add a new printer route to the routing file	A Add printer routes
Browse a printer route in the routing file	B Browse printer routes
Copy an old printer route to a new one	C Copy printer routes
Delete an existing printer route from the routing file	D Delete printer routes
Update an existing printer route	E Edit printer routes

In each of these tasks, you will be entering the MAJOR and MINOR NAME (key fields) of a new or existing record and then adding or updating data on a subsequent panel.

Table 10 on page 77 shows the data you need for these panels. See this table as you work with the routing file records.

Table 10. Routing file ISPF input fields

ISPF Field	Definition
MAJOR NAME	An 8-byte alphanumeric field containing the major name for this routing. For JES, this corresponds to the DEST parameter in the JCL. For VTAM, this is the logical printer LU name.
MINOR NAME	An 8-byte alphanumeric field containing the minor name for this routing. For JES, this corresponds to the CLASS and FORMS parameters in the JCL, 1 character for CLASS and 1 to 7 characters for FORMS. For VTAM, this is any value of your choice.
NO OF DEST	The number of destinations in this routing. The number 1 indicates a normal routing to one printer. A number greater than 1 indicates a specific broadcast routing to multiple homogenous printers.
RETAIN TIME(S)	The period of time the system should retain data for this routing after a successful transmission. The format is **dddhhmm** where: **ddd** = the number of days (range 000 to 366) **hh** = the number of hours (range 00 to 23) **mm** = the number of minutes (range 00 to 59)
RETAIN TIME(U)	The period of time the system should retain data for this routing after an unsuccessful transmission after the retry attempts have been exhausted. The format is **dddhhmm** where: **ddd** = the number of days (range 000 to 366) **hh** = the number of hours (range 00 to 23) **mm** = the number of minutes (range 00 to 59)
RETRY INTERVAL	The period of time the system should wait before attempting a retry on this routing after an unsuccessful attempt to transmit. The format is **dddhhmm** where: **ddd** = the number of days (range 000 to 366) **hh** = the number of hours (range 00 to 23) **mm** = the number of minutes (range 00 to 59)
RETRY LIMIT	The number of retry attempts to be made on this routing. Valid values are 0 to 65535. If the file is not successfully transmitted by LPR within this number of retries, the RETAIN TIME(U) becomes effective. If you specify 0, the print data and queue record will be deleted by the queue manager if the first transmission is not successful. 0 is the default.
OPTIONS NAME	The name of the options file record to be associated with this routing. This must be a valid name (key field) of an existing options file record.
ROUTING EXIT	The name of a specific routing exit to be invoked. See "Writing the specific routing exit" on page 97 for information.
HOST NAME/IP ADDR	The internet name or IP address of the destination host. This field can contain up to 255 characters and is case-sensitive. Use dotted decimal format for addresses and use periods to separate labels in a domain name. Be sure to specify the host in the same way on all routing records which specify the same printer. This is necessary to ensure correct FIFO (First In, First Out) handling of print jobs for that printer.

Table 10. Routing file ISPF input fields (continued)

ISPF Field	Definition
PRINTER NAME	The name of the printer to be used at the remote host for the destination. his field can contain up to 255 characters and is case-sensitive.
LUCLASS	For VTAM only, this is a list of printer classes assigned to this destination. It identifies which VTAM LUs are used by each occurrence of the VTAM capture point. Valid values are 1 to 64. Specify them from left to right with a space between.
DEF PAGE FORMAT	For VTAM only, the name of an entry in the page format table. The default is blank. See "Page format definition" on page 32 for information about creating the page format table.
EOFILE NAME	For VTAM only, the name of an entry in the end-of-file rules table. The default is DFLTNTRY. See "Defining end-of-file rules" on page 27 for information about creating the end-of-file table.

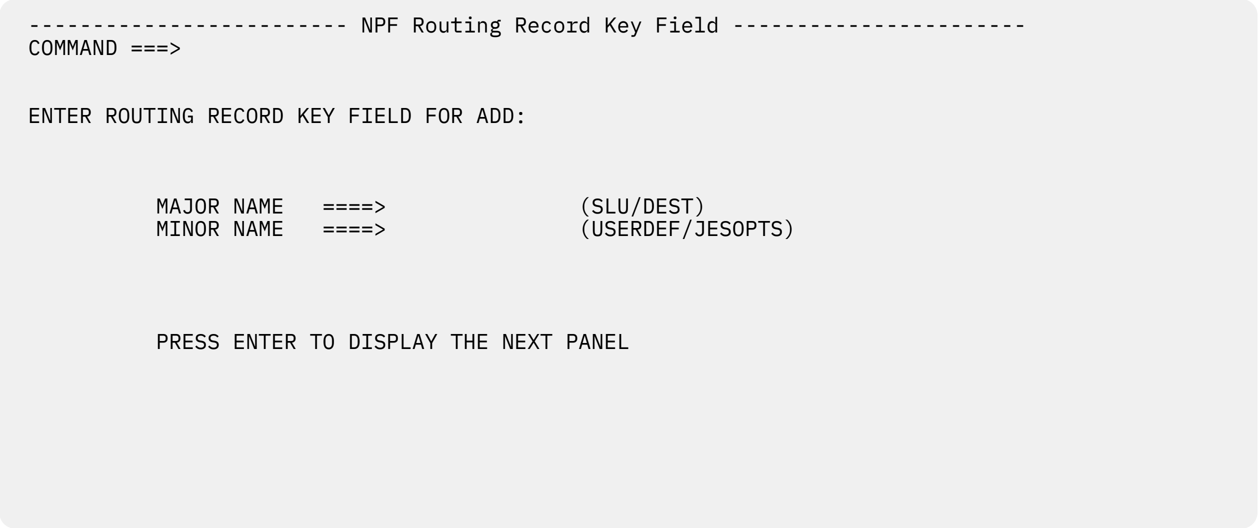
Adding records to the routing file

You can add routing records by creating a new record (new key fields and new data) or copying a new record (new key fields and new data). Whether you are adding a normal routing with one destination or a specific broadcast routing with multiple destinations, you begin in the same way.

Procedure

Follow these steps to add a new routing record:

1. Select **A** from the NPF Printer Routing Configuration panel to display the NPF Routing Record Key Field panel (Figure 26 on page 78)



```
------------------------ NPF Routing Record Key Field ----------------------
COMMAND ===>


ENTER ROUTING RECORD KEY FIELD FOR ADD:



          MAJOR NAME   ====>              (SLU/DEST)
          MINOR NAME   ====>              (USERDEF/JESOPTS)




          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 26. NPF routing file record key field panel (EZAPPN9)

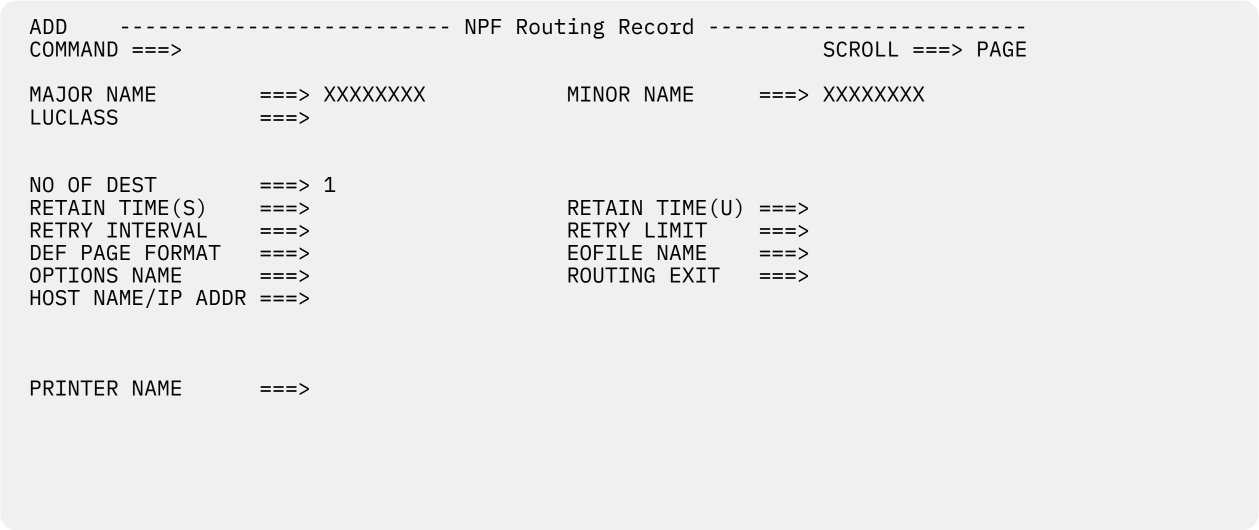
2. Specify the routing record key field. These are unique identifiers consisting of a MAJOR NAME and MINOR NAME. Enter the MAJOR NAME and MINOR NAME according to the requirements for a JES or VTAM configuration.

   The MAJOR NAME for JES corresponds to the JCL DEST parameter. For VTAM it is the logical printer LU name.

   The MINOR NAME for JES corresponds to the JCL CLASS and FORMS parameters, 1 character for CLASS and 1 to 7 characters for FORMS. Concatenate the two values and left justify them. For example, if CLASS=C and FORMS=1885, enter C1185 starting at the left of the MINOR NAME entry field.

For VTAM, the MINOR NAME is user-defined. It can contain any 1 to 8 alphanumeric characters of your choice.

3. Press **Enter** to display the NPF Routing Record Add panel (Figure 27 on page 79)



```
ADD    -------------------------- NPF Routing Record -------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

MAJOR NAME        ===> XXXXXXXX           MINOR NAME     ===> XXXXXXXX
LUCLASS           ===>


NO OF DEST        ===> 1
RETAIN TIME(S)    ===>                    RETAIN TIME(U) ===>
RETRY INTERVAL    ===>                    RETRY LIMIT    ===>
DEF PAGE FORMAT   ===>                    EOFILE NAME    ===>
OPTIONS NAME      ===>                    ROUTING EXIT   ===>
HOST NAME/IP ADDR ===>



PRINTER NAME      ===>
```

Figure 27. NPF routing record panel (EZAPPN15)

4. Specify the fields for the new routing you are adding. If this is a normal routing, you need only to enter data on this panel. If this is a specific broadcast routing, enter the information for the first destination on this panel and the information for the other destinations on subsequent panels. See Table 10 on page 77 for specific input criteria.
5. To save the data:
   - For a normal routing, press **ENTER** and then press the END PF key or enter **END** on the command line to return.
   - For a specific broadcast routing, press **ENTER** and then press the DOWN PF key to scroll forward to the next panel to enter data for the other printer destinations. When you are done, press **ENTER** and the END PF key to save the data for this destination.
   - For all panels except the first panel, you can also press the UP PF key to scroll backward to edit any data for these destinations.
6. To return without saving the routing, enter **CANcel** on the command line.

Results

Of particular note:

- MAJOR NAME and MINOR NAME are the key fields for this record and cannot be changed. If they are incorrect, you must cancel this record and start over.
- NO OF DEST specifies thenumber of destinations on the routing and determines if this is a normal or specific broadcast routing. Valid values are 1 to 65535. Normal routings must have 1 destination. This is the default.
- LUCLASS, DEF PAGE FORMAT, and EOFILE NAME are for VTAM routes only. Leave these fields blank for JES routes.
- OPTIONS NAME is required and must be the valid name (key field) of an existing options file record.
- HOST NAME/IP ADDR (Host Name or IP Address) and PRINTER NAME are required. They are both case sensitive and can contain uppercase and lowercase letters. Imbedded blanks are not allowed.

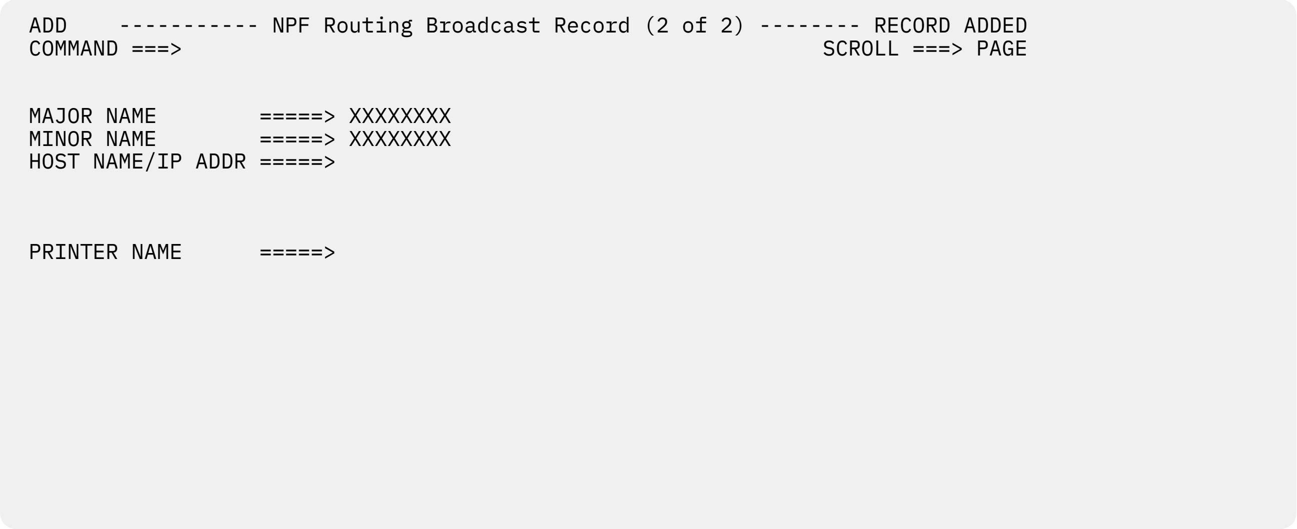
- If this is a specific broadcast routing (NO OF DEST is greater than 1), enter the HOST NAME/IP ADDR (Host Name or IP Address) and PRINTER NAME for the first destination on this panel.
- RETAIN TIME(S), RETAIN TIME(U), RETRY TIME, and RETRY LIMIT affect whether or not the print data is retained after the first transmission attempt and, if retained, how and when the system will try to resend it. See "Managing the print queues" on page 85 for more information.

Adding specific broadcast destinations

About this task

If you specify more than one destination on the Routing Record Add Panel (NO OF DEST is greater than 1), it indicates that you are creating a specific broadcast routing to send the print data to multiple homogenous printers.

When you scroll forward, the system will display subsequent panels so you can enter information for the other destinations. NPF provides one Routing Broadcast Record panel (Figure 28 on page 80) for each additional destination.



```
ADD    ----------- NPF Routing Broadcast Record (2 of 2) -------- RECORD ADDED
COMMAND ===>                                                  SCROLL ===> PAGE


MAJOR NAME        =====> XXXXXXXX
MINOR NAME        =====> XXXXXXXX
HOST NAME/IP ADDR =====>



PRINTER NAME      =====>
```

Figure 28. NPF routing broadcast record panel (EZAPPN23)

Procedure

Follow these steps to add specific broadcast destinations:

1. Complete the HOST NAME/IP ADDR (Host Name or IP Address) and PRINTER NAME for the next destination on the routing and press **ENTER** to save the data for this destination.
2. Press the **DOWN** PF key to scroll forward. Subsequent panels will appear for as many additional destinations as are needed. For example, if the number of destinations is 5, then this panel will appear 4 times to enable you to complete information about each specific destination.
3. By pressing the **UP** PF key, you can also scroll backward to edit any data for these destinations.
4. When you are done, press the END PF key or enter **END** on the command line to return.

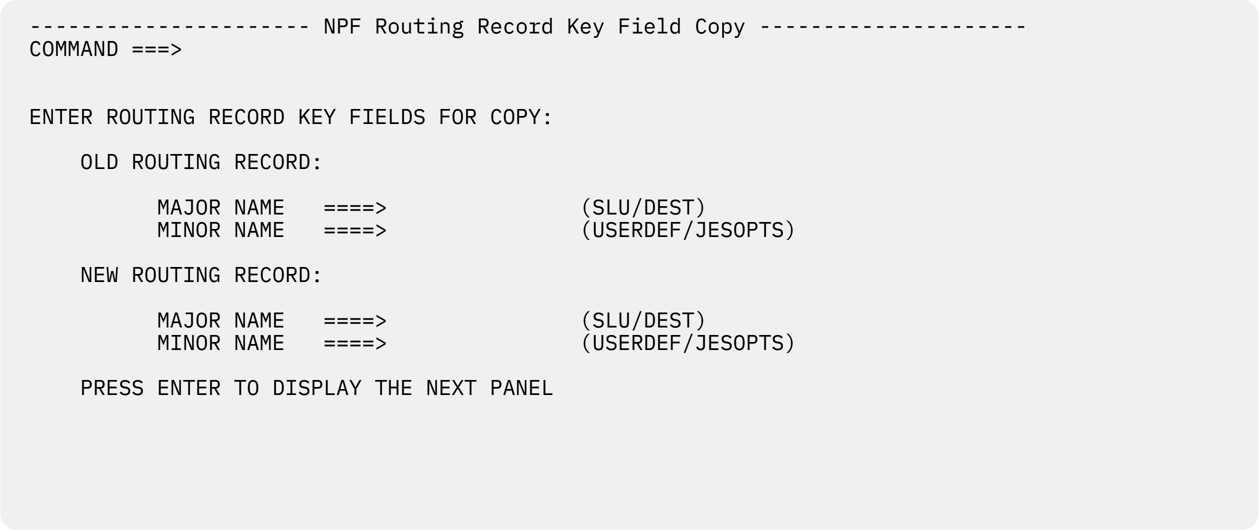
Copying records in the routing file

You can create a new routing file record by copying data from an existing record to a new record. If the existing routing is a specific broadcast, all the destinations will be copied. Because most routing records are similar except for minor variations, this feature can save time.

Procedure

Follow these steps to copy a routing file record from an existing record:

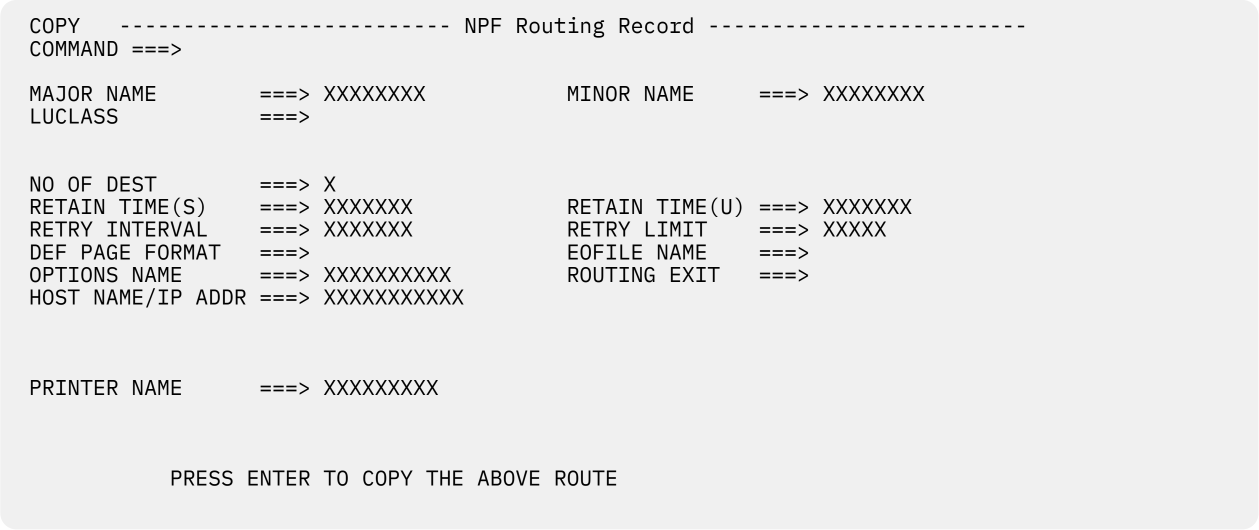
1. Select **C** from the NPF Printer Routing Configuration panel to display the NPF Routing Record Key Field Copy panel (Figure 29 on page 81)



```
---------------------- NPF Routing Record Key Field Copy ---------------------
COMMAND ===>


ENTER ROUTING RECORD KEY FIELDS FOR COPY:

    OLD ROUTING RECORD:

          MAJOR NAME   ====>               (SLU/DEST)
          MINOR NAME   ====>               (USERDEF/JESOPTS)

    NEW ROUTING RECORD:

          MAJOR NAME   ====>               (SLU/DEST)
          MINOR NAME   ====>               (USERDEF/JESOPTS)

    PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 29. NPF routing file record key field copy panel (EZAPPN12)

2. Specify the key fields for OLD ROUTING RECORD and the NEW ROUTING RECORD. Enter the MAJOR NAME and MINOR NAME of the existing record you are copying in the OLD ROUTING RECORD field. Enter the MAJOR NAME and MINOR NAME of the record you are creating in the NEW ROUTING RECORD field. This will be the key field for the new routing record.
3. Press **Enter** to display the NPF Routing Record Panel (Figure 30 on page 81.)



```
COPY   -------------------------- NPF Routing Record -------------------------
COMMAND ===>

MAJOR NAME        ===> XXXXXXXX           MINOR NAME     ===> XXXXXXXX
LUCLASS           ===>


NO OF DEST        ===> X
RETAIN TIME(S)    ===> XXXXXXX            RETAIN TIME(U) ===> XXXXXXX
RETRY INTERVAL    ===> XXXXXXX            RETRY LIMIT    ===> XXXXX
DEF PAGE FORMAT   ===>                    EOFILE NAME    ===>
OPTIONS NAME      ===> XXXXXXXXXX         ROUTING EXIT   ===>
HOST NAME/IP ADDR ===> XXXXXXXXXXX



PRINTER NAME      ===> XXXXXXXXX



           PRESS ENTER TO COPY THE ABOVE ROUTE
```

Figure 30. NPF routing record panel (EZAPPN15)

This panel displays the data to be copied from the existing record. The information on this panel cannot be changed.

4. Press **Enter** to copy the routing file record.

Copying specific broadcast destinations

You can copy only an entire specific broadcast routing, not individual destinations within it. When you copy the normal record for the first destination on a specific broadcast routing, all the records for the other destinations are copied with it.

To copy a specific broadcast routing, follow the steps for copying a normal routing file record. See "Copying records in the routing file" on page 80. If this is a routing with multiple destinations, the records for all the additional destinations will be copied along with the first one.

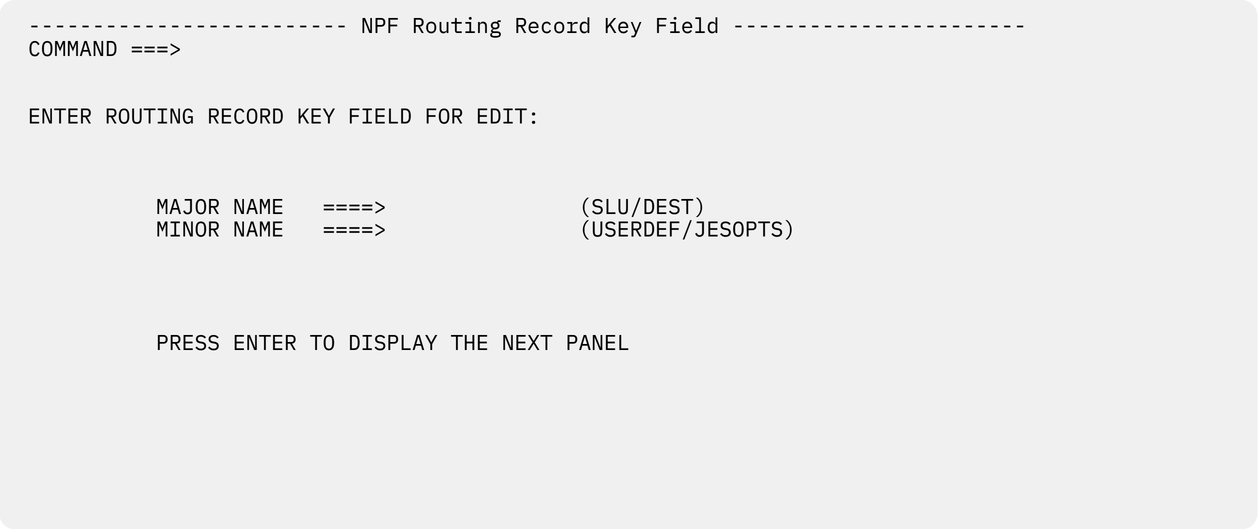
Updating records in the routing file

To update any existing routing file record, you must know the MAJOR NAME and MINOR NAME of the routing you want to update. Use the same rules for updating data as for entering new data in these fields.

Procedure

Follow these steps to update a routing record:

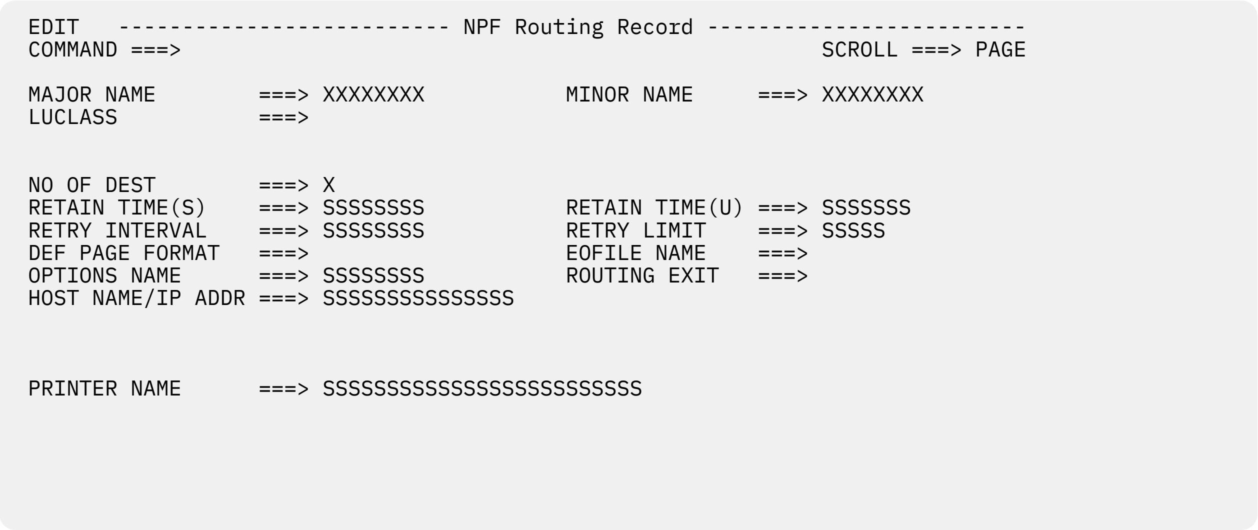
1. Select **E** from the NPF Printer Routing Configuration panel to display the NPF Routing Record Key Field Edit panel (Figure 31 on page 82)



```
------------------------- NPF Routing Record Key Field ------------------------
COMMAND ===>


ENTER ROUTING RECORD KEY FIELD FOR EDIT:



          MAJOR NAME   ====>               (SLU/DEST)
          MINOR NAME   ====>               (USERDEF/JESOPTS)




          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 31. NPF routing record key field edit panel (EZAPPN9)

2. Specify the ROUTING RECORD KEY FIELD. Enter the MAJOR NAME and MINOR NAME for the record you want to update.
3. Press **Enter** to display the NPF Routing Records Edit panel (Figure 32 on page 83)



```
EDIT   ------------------------- NPF Routing Record -------------------------
COMMAND ===>                                                  SCROLL ===> PAGE

MAJOR NAME        ===> XXXXXXXX           MINOR NAME     ===> XXXXXXXX
LUCLASS           ===>


NO OF DEST        ===> X
RETAIN TIME(S)    ===> SSSSSSSS           RETAIN TIME(U) ===> SSSSSSS
RETRY INTERVAL    ===> SSSSSSSS           RETRY LIMIT    ===> SSSSS
DEF PAGE FORMAT   ===>                    EOFILE NAME    ===>
OPTIONS NAME      ===> SSSSSSSS           ROUTING EXIT   ===>
HOST NAME/IP ADDR ===> SSSSSSSSSSSSSSS



PRINTER NAME      ===> SSSSSSSSSSSSSSSSSSSSSSSSS
```

Figure 32. NPF routing record edit panel (EZAPPN15)

4. Change the information as necessary to update the record. See Table 10 on page 77 for specific input criteria and to “Adding records to the routing file” on page 78 for helpful notes.
5. Press **Enter** to save the new values.

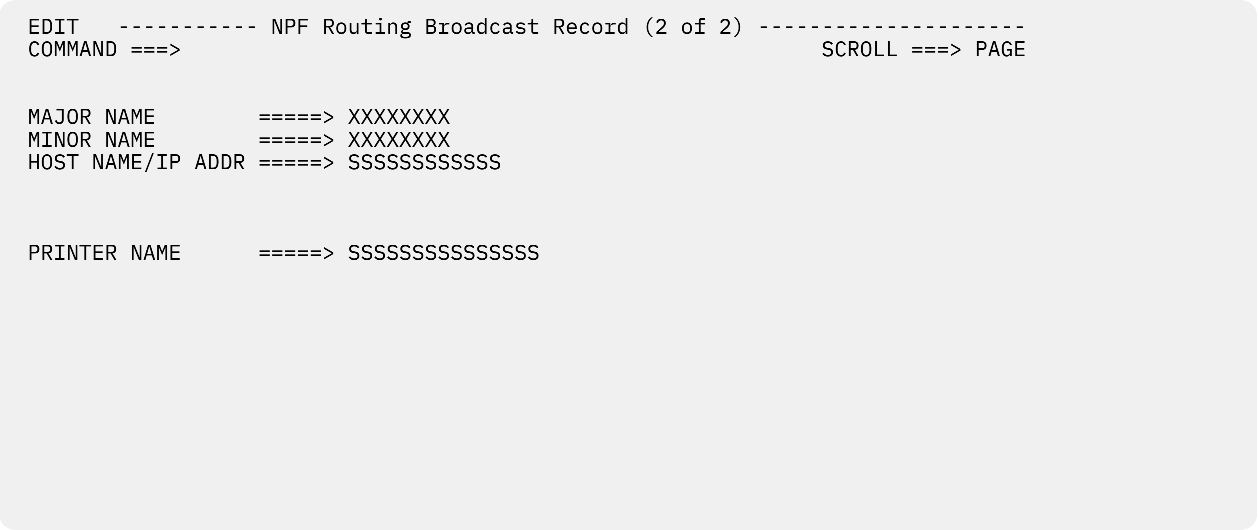
Updating specific broadcast destinations

To update any information for a specific broadcast routing, start by displaying the normal record for the first destination and then scroll through the subsequent records for each of the other destinations.

Procedure

Follow these steps to update a specific routing record:

1. Select **E** from the NPF Printer Routing Configuration panel to display the NPF Routing Record Key Field Edit panel (Figure 31 on page 82).
2. Specify the ROUTING RECORD KEY FIELD. Enter the MAJOR NAME and MINOR NAME for the record you want to update.
3. Press **Enter** to display the NPF Routing Records Edit panel (Figure 32 on page 83).
4. Change any data as necessary on this panel. Press **Enter** to save any changes to the first destination.
5. Press the **DOWN** or **UP** PF keys to scroll forward or backward to display the data for each destination. (See Figure 33 on page 84.)



```
EDIT   ----------- NPF Routing Broadcast Record (2 of 2) ---------------------
COMMAND ===>                                                  SCROLL ===> PAGE


MAJOR NAME        =====> XXXXXXXX
MINOR NAME        =====> XXXXXXXX
HOST NAME/IP ADDR =====> SSSSSSSSSSSS



PRINTER NAME      =====> SSSSSSSSSSSSSSS
```

Figure 33. NPF routing broadcast record edit panel (EZAPPN23)

6. Change the HOST NAME/IP ADDR (Host Name or IP Address) and PRINTER NAME, if necessary, for the next destination. Then, scroll forward to the next destination.
7. Repeat the last step until the panels for all of the destinations have been displayed and updated or bypassed.

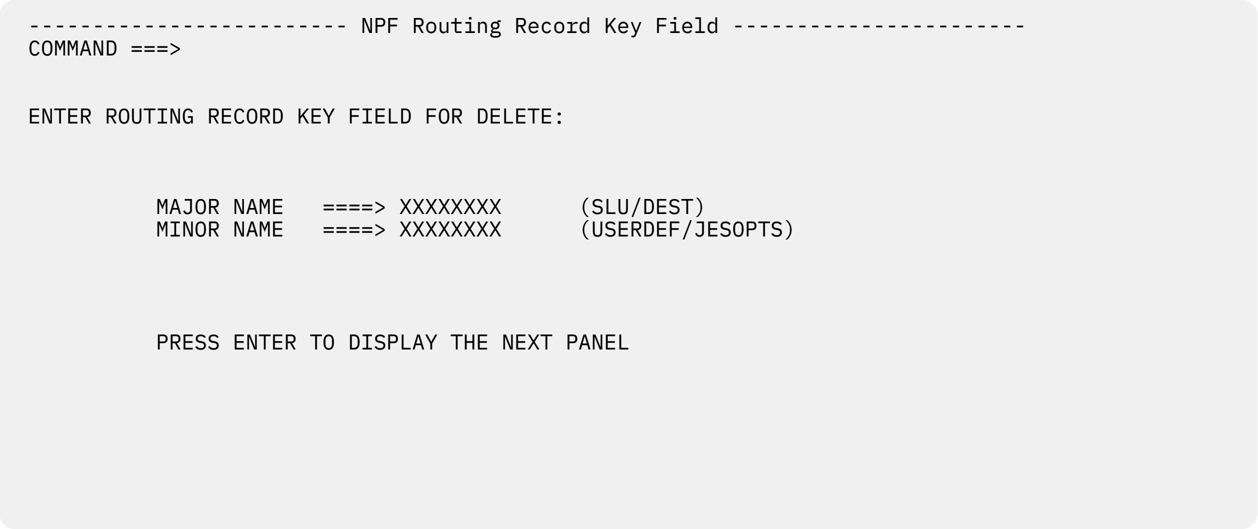
Deleting records from the routing file

You can delete records in the routing file for normal or specific broadcast routings. Be aware that when you delete these records, you are eliminating the routing from your system.

Procedure

Follow these steps to delete a routing record:

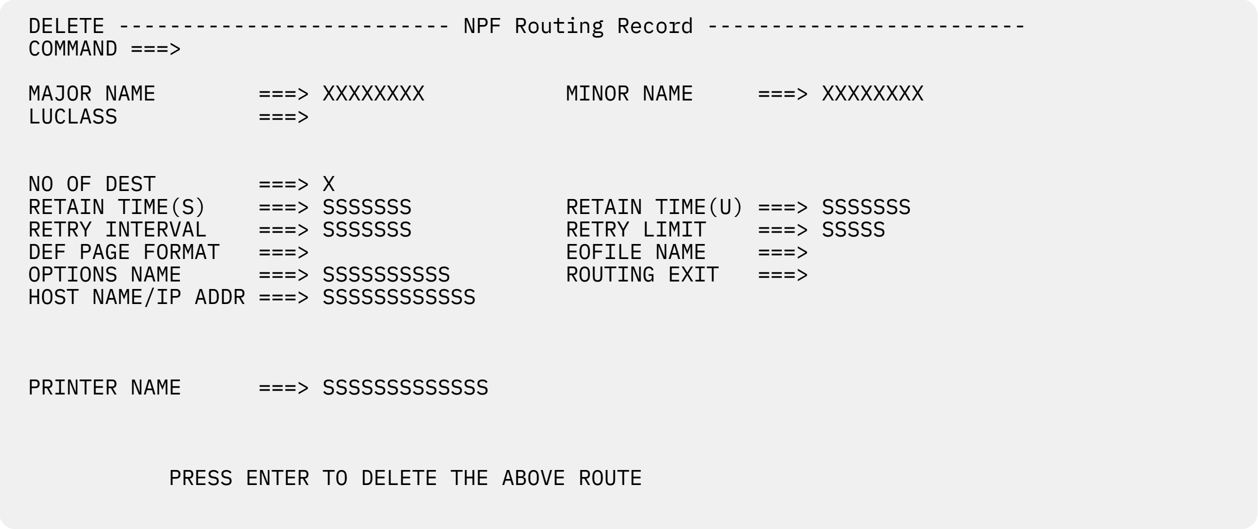
1. Select **D** from the NPF Printer Routing Configuration panel to display the NPF Routing Record Key Field Delete panel (Figure 31 on page 82).
2. Specify the ROUTING RECORD KEY FIELD. Enter the MAJOR NAME and MINOR NAME for the record you want to update.
3. Press **Enter** to display the NPF Routing Records Delete panel (Figure 34 on page 84).



```
------------------------- NPF Routing Record Key Field -----------------------
COMMAND ===>


ENTER ROUTING RECORD KEY FIELD FOR DELETE:



          MAJOR NAME   ====> XXXXXXXX      (SLU/DEST)
          MINOR NAME   ====> XXXXXXXX      (USERDEF/JESOPTS)




          PRESS ENTER TO DISPLAY THE NEXT PANEL
```

Figure 34. Routing file record key field delete panel (EZAPPN9)

4. Specify the ROUTING RECORD KEY FIELD. Enter the MAJOR NAME and MINOR NAME for the record you want to delete.
5. Press **Enter** to display the NPF Routing Record Delete Panel (Figure 35 on page 85)



```
DELETE -------------------------- NPF Routing Record -------------------------
COMMAND ===>

MAJOR NAME         ===> XXXXXXXX            MINOR NAME     ===> XXXXXXXX
LUCLASS            ===>


NO OF DEST         ===> X
RETAIN TIME(S)     ===> SSSSSSS             RETAIN TIME(U) ===> SSSSSSS
RETRY INTERVAL     ===> SSSSSSS             RETRY LIMIT    ===> SSSSS
DEF PAGE FORMAT    ===>                     EOFILE NAME    ===>
OPTIONS NAME       ===> SSSSSSSSSS          ROUTING EXIT   ===>
HOST NAME/IP ADDR ===> SSSSSSSSSSSS



PRINTER NAME       ===> SSSSSSSSSSSSS



           PRESS ENTER TO DELETE THE ABOVE ROUTE
```

Figure 35. NPF routing record delete panel (EZAPPN15)

6. Verify that this is the record you want to delete.
7. Press **Enter** to delete it.

   If you requested delete confirmation for the routing file, you will get a panel that allows you to continue or stop the deletion. See "Setting the NPF ISPF defaults" on page 68 for instructions to turn delete confirmation on and off.

   Note: If you are deleting a specific broadcast routing, each destination on the routing will be deleted.

Deleting specific broadcast destinations

You can delete only an entire specific broadcast routing, not individual destinations within it. When you delete the normal record for the first destination on a specific broadcast routing, all the records for the other destinations are deleted with it.

To delete a specific broadcast routing, follow the steps for deleting a normal routing file record. See "Deleting records from the routing file" on page 84. If this is a routing with multiple destinations, the records for all the additional destinations will be deleted along with the first one.

Managing the print queues

Whenever a print job is received by NPF, the capture point application creates a temporary QSAM print data set and adds a record to the queue file for each destination to which that print data is to be sent. For a multiple-destination routing only, an extra record is also written to act as an end-of-set marker.

The queue manager then assumes responsibility for all further processing of the print data set and the queue file records which point to it. Queue manager functions are listed as follows:

- Determining the order in which print jobs should be scheduled. For any given printer, print jobs are sent in FIFO (First In, First Out) order based on arrival time.
- Sending print data to LPD, including both initial send attempts and retries.
- Evaluating the return code from the send attempt and determining which of the following actions is needed:
  - After a successful transmission, keeping the queue record for a user-specified retain time or erasing the queue record immediately.

  - After an unsuccessful transmission, setting up the queue record for a retry of the send, keeping the queue record for a user-specified retain time, or erasing the queue record immediately.
- Erasing a queue record after its user-specified retain time has expired.
- Deleting a QSAM print data set after the last queue file record which points to it has been erased.

Each record on the queue file contains a data set status field. Possible values for this field and their meanings are as follows:

Z

No send attempted yet. Normal next action is the initial attempt to send the print data.

U

Last send unsuccessful; one or more retries left. Normal next action is a retry of the failed send operation.

R

Last send unsuccessful; all retries exhausted; queue record being kept for the RETAIN(U) retain time. Normal next action, after the RETAIN(U) interval expires, is to erase the queue record and, if this is a single-destination routing, delete the corresponding print data set.

T

Last send successful; queue record being kept for the RETAIN(S) retain time. Normal next action, after the RETAIN(S) interval expires, is to erase the queue record and, if this is a single-destination routing, delete the corresponding print data set.

H

Print job being held. The queue record is kept, but no processing is done for it while in this state.

X

End-of-set marker for a multiple-destination routing. The purpose of this record is to make sure that the QSAM print data set is not deleted until all queue records which point to it have been erased. Records with this state are displayed by the ISPF interface, but you cannot make any changes to them.

D

Delete. Records with this state have been marked for deletion by a panel operator but not yet erased by the queue manager. These records are never displayed by the ISPF interface.

Working with the queue file

Queue file record data

Table 11 on page 86 shows the data you need to manage your print queue. See this table as you work with the queue file records.

Table 11. Queue file ISPF fields

ISPF Field	Definition
SLU/DEST	An 8-byte alphanumeric field containing the first part of the queue record key. For JES routes, this corresponds to the DEST parameter in the JCL. For VTAM, this is the secondary logical unit (logical printer LU name).
PLU/JOBNAME	An 8-byte alphanumeric field containing the second part of the queue record key. For JES routes, this corresponds to the jobname used to submit the print request. For VTAM, this is the primary logical unit (origin of the data).
Creation Date	The date the queue record was created. The format will vary according to the national language being used.
Creation Time	The time the queue record was created. The format will vary according to the national language being used.

Table 11. Queue file ISPF fields (continued)

ISPF Field	Definition
DS Status	A code representing the status of the print data: **Z** Successfully received by NPF but not yet sent to remote printer **U** Unsuccessful transmission; to be retried **R** Unsuccessful transmission; retries exhausted **T** Successful transmission **H** Being held in the print queue **X** End of Set of queue records
Printer Name	Same as PRINTER NAME in the routing file. The name of the printer used at the destination host on this route. This can be up to 255 characters and is case-sensitive.
Host Name/IP Address	Same as HOST/IP ADDR in the routing file. The name or IP address of the destination host on this route. destination host on this route. This can be up to 255 characters and is case-sensitive. Use dotted decimal format for addresses and periods to separate labels in a domain name. Embedded blanks are not allowed. Be sure to specify the host in the same way on all routing records which specify the same printer. This is necessary to ensure correct FIFO (First In, First Out) handling of print jobs for that printer.
DS Name	Name of the print data set. See name for details on how this name is created.
Last Send Date/Time	The last date and time the queue manager tried to send this print request. The format will vary according to the national language being used.
Next Send Date/Time	The next date and time the queue manager will try to send this print request. This is calculated by the last sent date/time and the Retry Interval values. The format will vary according to the national language being used.
Retry Interval	Same as RETRY TIME in the routing file. The period of time the system should wait before attempting a retry on this route after an unsuccessful attempt to transmit.
Attempts/of	The number of times the queue manager has actually tried to resend the print request and the number of total retry attempts that were specified. The second value is the same as RETRY LIMIT in the routing file.
Retain Time (Successful)	The amount of time the queue manager should keep the print data after successful transmission. When this time is exceeded, the record is deleted.
Retain Time (Unsuccessful)	The amount of time the queue manager should keep the print data after an unsuccessful transmission when there are no more retries scheduled.
Print Options	Same as LPR Options in the options file. A list of LPR options that will be used to print the data at the destination printer.

Displaying the queue file

Depending on the size of your installation and the volume of requests, the queue file can contain numerous records. You can display the entire queue file or just a selected part of it. You can choose to select queue records based on the key fields, the creation data and time, the data set status, the printer queue name, host name, or IP address.

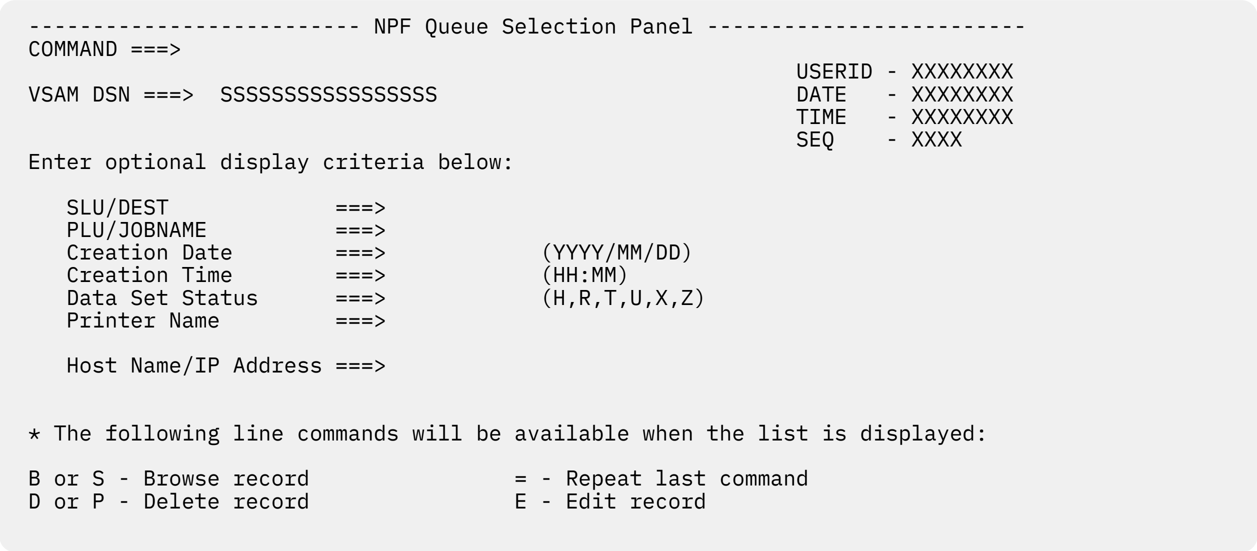
Procedure

Follow these steps to display the queue file:

1. Select **Q** from the Network Print Facility Primary Option Menu to display the NPF Queue Selection Panel (Figure 36 on page 88)

   This panel provides entry fields for your selection criteria and shows you the commands you are authorized to use on the queue records listed in the subsequent panel. It also displays the queue file (VSAM DSN) you will access.

   The file displayed is either the default file or the last file you used. If you want to work with a different queue file, enter a fully-qualified data set name in this field. If you want to reset this field to the default, clear the field and press **ENTER**.



```
-------------------------- NPF Queue Selection Panel -------------------------
COMMAND ===>
                                                            USERID - XXXXXXXX
VSAM DSN ===>  SSSSSSSSSSSSSSSSS                            DATE   - XXXXXXXX
                                                            TIME   - XXXXXXXX
                                                            SEQ    - XXXX
Enter optional display criteria below:

   SLU/DEST             ===>
   PLU/JOBNAME          ===>
   Creation Date        ===>            (YYYY/MM/DD)
   Creation Time        ===>            (HH:MM)
   Data Set Status      ===>            (H,R,T,U,X,Z)
   Printer Name         ===>

   Host Name/IP Address ===>


* The following line commands will be available when the list is displayed:

B or S - Browse record                = - Repeat last command
D or P - Delete record                E - Edit record
```

Figure 36. NPF queue selection panel (EZAPPQSP)

2. Specify the fields to select the records you want to see. The values in these fields are treated in an ***AND*** condition. The selected records must meet all the criteria specified.

   You can use an asterisk (*) in the following ways:

   - By itself to mean any value (* matches any value)
   - At the start of a string (*ABC matches values ending with ABC)
   - At the end of a string (ABC* matches values starting with ABC)

   Although the fields for Printer Name and Host Name or IP Address contain up to 255 characters, you can enter only 130 characters. If necessary, you can use the asterisk (*) to accommodate the selection of longer names.

   If all fields are blank or contain an asterisk (*), you will get a list of the entire NPF queue file.

3. If necessary, change the SEQ field value to indicate the order in which you want the selected queue records displayed. You have two valid choices:

   - KEY - displays the records in key sequence, the same order in which the records occur within the queue file.

- TIME - causes the records to be displayed in arrival-time sequence, with the oldest records appearing first.

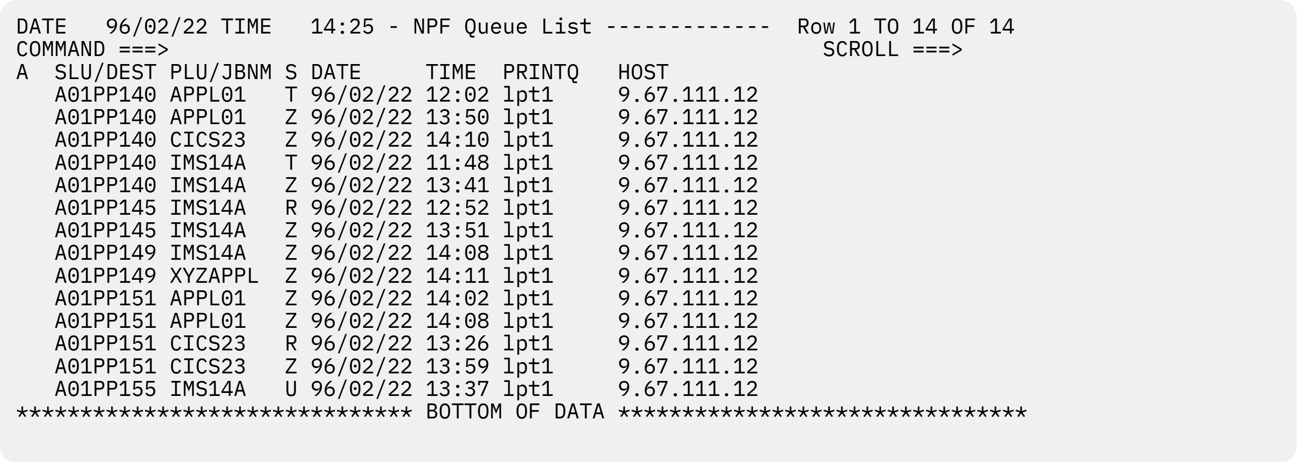
4. Press **Enter** to display the NPF Queue List panel (Figure 37 on page 89).

Selecting a queue record

About this task

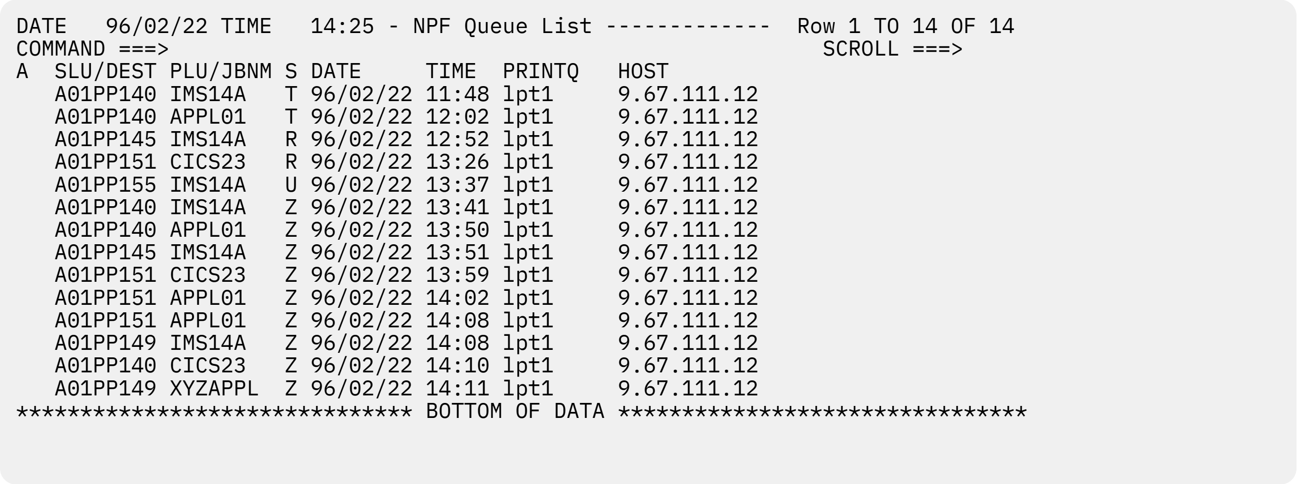
The records displayed in the NPF Queue List panel are selected based on the criteria you enter on the NPF Queue Selection panel. You can view these queue records, and if authorized, modify, delete, reset, or hold them.

Note: The examples below show sample queue records for printer lpt1 at IP address 9.67.111.12 arranged in both KEY and TIME sequences.



```
DATE   96/02/22 TIME   14:25 - NPF Queue List -------------  Row 1 TO 14 OF 14
COMMAND ===>                                                   SCROLL ===>
A  SLU/DEST PLU/JBNM S DATE     TIME  PRINTQ   HOST
   A01PP140 APPL01   T 96/02/22 12:02 lpt1     9.67.111.12
   A01PP140 APPL01   Z 96/02/22 13:50 lpt1     9.67.111.12
   A01PP140 CICS23   Z 96/02/22 14:10 lpt1     9.67.111.12
   A01PP140 IMS14A   T 96/02/22 11:48 lpt1     9.67.111.12
   A01PP140 IMS14A   Z 96/02/22 13:41 lpt1     9.67.111.12
   A01PP145 IMS14A   R 96/02/22 12:52 lpt1     9.67.111.12
   A01PP145 IMS14A   Z 96/02/22 13:51 lpt1     9.67.111.12
   A01PP149 IMS14A   Z 96/02/22 14:08 lpt1     9.67.111.12
   A01PP149 XYZAPPL  Z 96/02/22 14:11 lpt1     9.67.111.12
   A01PP151 APPL01   Z 96/02/22 14:02 lpt1     9.67.111.12
   A01PP151 APPL01   Z 96/02/22 14:08 lpt1     9.67.111.12
   A01PP151 CICS23   R 96/02/22 13:26 lpt1     9.67.111.12
   A01PP151 CICS23   Z 96/02/22 13:59 lpt1     9.67.111.12
   A01PP155 IMS14A   U 96/02/22 13:37 lpt1     9.67.111.12
******************************* BOTTOM OF DATA ********************************
```

Figure 37. NPF queue list (EZAPPQLP) in KEY sequence



```
DATE   96/02/22 TIME   14:25 - NPF Queue List -------------  Row 1 TO 14 OF 14
COMMAND ===>                                                   SCROLL ===>
A  SLU/DEST PLU/JBNM S DATE     TIME  PRINTQ   HOST
   A01PP140 IMS14A   T 96/02/22 11:48 lpt1     9.67.111.12
   A01PP140 APPL01   T 96/02/22 12:02 lpt1     9.67.111.12
   A01PP145 IMS14A   R 96/02/22 12:52 lpt1     9.67.111.12
   A01PP151 CICS23   R 96/02/22 13:26 lpt1     9.67.111.12
   A01PP155 IMS14A   U 96/02/22 13:37 lpt1     9.67.111.12
   A01PP140 IMS14A   Z 96/02/22 13:41 lpt1     9.67.111.12
   A01PP140 APPL01   Z 96/02/22 13:50 lpt1     9.67.111.12
   A01PP145 IMS14A   Z 96/02/22 13:51 lpt1     9.67.111.12
   A01PP151 CICS23   Z 96/02/22 13:59 lpt1     9.67.111.12
   A01PP151 APPL01   Z 96/02/22 14:02 lpt1     9.67.111.12
   A01PP151 APPL01   Z 96/02/22 14:08 lpt1     9.67.111.12
   A01PP149 IMS14A   Z 96/02/22 14:08 lpt1     9.67.111.12
   A01PP140 CICS23   Z 96/02/22 14:10 lpt1     9.67.111.12
   A01PP149 XYZAPPL  Z 96/02/22 14:11 lpt1     9.67.111.12
******************************* BOTTOM OF DATA ********************************
```

Figure 38. NPF queue list (EZAPPQLP) in TIME sequence

Procedure

Follow these steps to select a record from this list:

1. Enter an action or command:

- To browse a record, enter **B** or **S** in the Action column. When the record is displayed, all the fields are protected.
- To delete a record, enter **D** or **P** in the Action column. The queue manager deletes the queue record and any of its print data sets.
- To edit a record, enter **E** in the Action column. When the record is displayed, you are able to modify fields not under system control.
- To hold a record, enter **H** in the Action column or **HOLD *slu/destname*** on the command line.
- To reset a record, enter **R** in the Action column or **RESET *slu/destname*** on the command line.
- To repeat any command, enter **=** in the Action column. This allows you to perform the same command successively for several records.
- To refresh the queue list display, enter **REFRESH** on the command line.

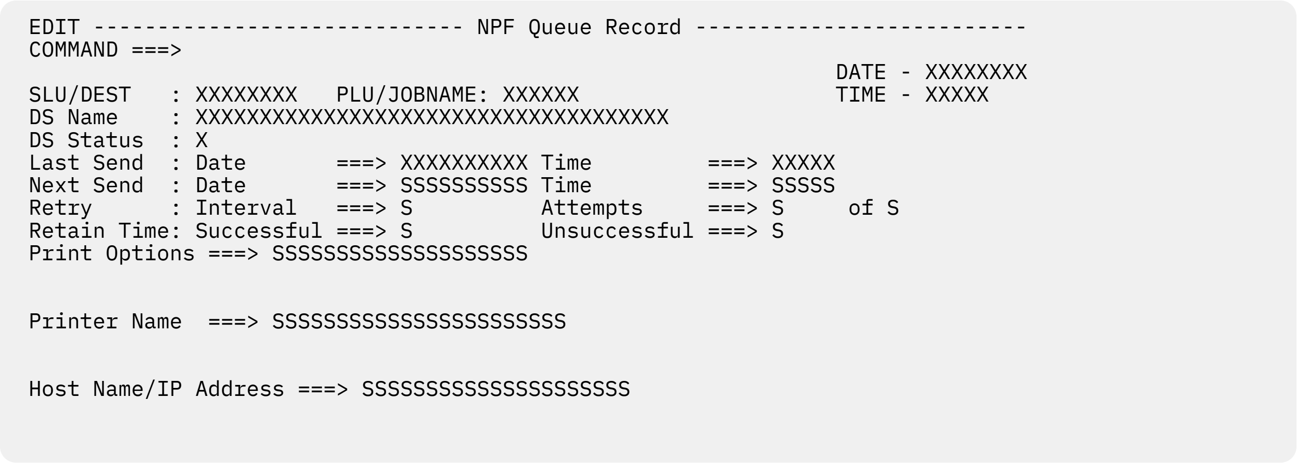
2. Press **Enter** to activate the commands and display the NPF Queue Record panel (Figure 39 on page 91).

Helpful hints

- You can reset queue records whose Data Set Status is U (unsuccessful), R (retries exhausted), or H (on hold). These jobs will be resent to the printer.
  - The reset action (**R**) resets one print request.
  - The **RESET *slu/destname*** command resets all requests for a destination.
- You can hold queue records whose Dataset Status is U (unsuccessful), R (retries exhausted), or Z (not yet sent to the printer). These jobs will not be sent to the printer.
  - The hold action (**H**) holds one print request.
  - The **HOLD *slu/destname*** command resets all requests for a destination.
- The queue records with a Dataset Status of X (End of Set) are only for system use. You cannot perform any functions on them.
- The DATE and TIME displayed on the Queue List panel are close to but not identical to the arrival date/time used to determine TIME ordering. This difference can sometimes make a list appear slightly out of order, even though the sequence is actually correct.
  - The displayed values indicate when the QSAM print data set was OPENed.
  - The arrival date/time values indicate when the QSAM print data set was CLOSEd.
- There are at least two cases in which the TIME sequencing of displayed queue file records should be useful:
  - To quickly check your network for printers experiencing delays, set the Data Set Status selection field to either UZ or ZU and set the SEQ selection field to TIME. Queue records for the oldest unsent print jobs will appear at the top of the resulting list, indicating which printers have the longest delays.
  - To display the backlog of unsent print jobs for a specific printer, set the Host Name/IP Address (Host Name or IP Address) and Printer Name fields to select that printer, and set other selection fields the same as in the previous example. The resulting list will show the waiting print jobs in the exact order in which they are scheduled to be printed.

Changing values in a queue record

You can change the values in the fields on this panel to affect when (and if) the queue manager will retry to transmit the job and when it will delete the queue record and its print data sets. You can also reroute the data on the next retry to a different destination and change the print options that will be used.



```
EDIT ----------------------------- NPF Queue Record -------------------------
COMMAND ===>
                                                               DATE - XXXXXXXX
SLU/DEST   : XXXXXXXX   PLU/JOBNAME: XXXXXX                    TIME - XXXXX
DS Name    : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
DS Status  : X
Last Send  : Date       ===> XXXXXXXXXX Time         ===> XXXXX
Next Send  : Date       ===> SSSSSSSSSS Time         ===> SSSSS
Retry      : Interval   ===> S          Attempts     ===> S     of S
Retain Time: Successful ===> S          Unsuccessful ===> S
Print Options ===> SSSSSSSSSSSSSSSSSSSS


Printer Name  ===> SSSSSSSSSSSSSSSSSSSSSSS


Host Name/IP Address ===> SSSSSSSSSSSSSSSSSSSSS
```

Figure 39. NPF queue record (EZAPPQRP)

Change data as appropriate in the following fields:

Next Send Date/Time
Retry Interval
Attempts/of (Corresponds to retry count and retry limit)
Retain Time Successful
Retain Time Unsuccessful
Print Options
Printer Name
Host Name or IP Address

See "Managing the print queues" on page 85 and Table 11 on page 86 for valid values for information on these fields.

Helpful hints

- To retry printing, increase the retry limit or decrease the retry count.

  For a queue record currently in state "R" (retries exhausted), either of these actions will change the state to "U" (unsuccessful).

  Attention: To avoid unnecessary print delays, you must modify the Next Send time also; the state change does not automatically update the Next Send time.

  Suggestion: A safer way to retry printing is to use the RESET command to change the record state and automatically update the Next Send time.
- To print the job with different options, change the Print Options.
- To delay or expedite additional print attempts, change the Next Send times and, possibly, the Retry Interval.
- To stop additional print attempts, change the retry limit to 0.
- To retain the queue record for a longer or shorter time, change the Retain Time values.
- To send the print job to another print queue or host, change the Printer Name, or Host Name or IP Address

Deleting queue records

You can delete queue records by entering **D** or **P** in the Action column on the NPF Queue List panel (Figure 37 on page 89).

Chapter 7. Writing exit routines to tailor the Network Print Facility

Installations can create exit routines to preclude the use of the Network Print Facility's own mechanism for routing by:

- Creating the routing data area (RDA)
- Modifying the key (for JES, DEST and CLASS, for example) and access the routing and options files
- Modifying the RDA after it has been created
- Modifying print data

Such exit routines must supply all the information the Network Print Facility requires involving routing and options. These optional exit routines can route output dynamically by modifying the routing data area (RDA), a critical data area described in more detail in "The Routing Data Area (RDA)" on page 108.

See Table 12 on page 93 for a short description of the Network Print Facility exits.

Table 12. Network Print Facility user exits

Exit	Description
Routing Exits - alter the routing of printed output • Used to determine where output is printed. (Substitute for the routing file or options file, or both.) • Used to modify routing file information. • Used to modify options file information.	Two types: 1. ***General routing exit*** - changes JES and VTAM input parameters ***before*** the routing or options data is read. It can replace the functions of the routing or options files or change the key used to access the routing or options files. 2. ***Specific routing exit*** - invoked ***after*** the routing or options data is read or ***after*** the general routing exit completes. It can change or augment the routing and options data obtained from those sources.
Input record exit - alters the content of the data to be printed. • Invoked each time data is received by the Network Print Facility. • Used to insert, modify, or delete application data. • Used to modify routing or options data based on contents of application data.	Allows specification of reformatting, translation, encryption, or other installation-defined functions needed to accommodate output to destined printers.

The exits are entered through an MVS link and follow normal MVS register conventions.

This chapter presents the following information:

- General routing exit information
- Specific routing exit information
- Input record exit information
- The routing data area
- Using the EZAPPFCD macro to generate the data areas you need

Two routing exits and one input record exit

Two separate user exits can be used to alter routing. These are referred to as *routing exits.* They substitute for the routing file or the options file or both.

- ***General routing exit***

  This user exit is named EZAPPGPR. It changes JES or VTAM input parameters and is invoked ***before*** the routing or options data is read (routing data areas are created.) It can replace the functions of the routing or options files because you can use it to create the RDA itself or change the key used to access the files. To invoke EZAPPGPR, it must be in a library specified by a JOBLIB, STEPLIB, or in the LNKLIST.

- ***Specific routing exit***

  This user exit is user-named and is invoked in one of two ways:

  1. ***After*** the general routing exit completes
  2. ***After*** the Network Print Facility creates the routing data area

  It can modify the RDA or change or augment the data obtained from those sources. To invoke the specific routing exit, it must be a LINKLST member of MVS PARMLIB or specified on either the JOBLIB or STEPLIB DD statements.

  The name of this exit is user-dependent. It must follow MVS module naming standards, can be up to an 8-byte alphanumeric name and can be defined in the:

  – RTDASREX field of the routing data area, via the general routing exit.

  – SREXIT parameter in "EZAPPFL TYPE=ROUTING" on page 45.

  – ROUTING EXIT field, as illustrated in Figure 27 on page 79.

  Input record exit

  This user exit performs two functions:

  1. Allows modification of the RDA.
  2. Allows the alteration of the contents of data to be printed. This can be used to insert, modify, or delete application data as needed to accommodate output to destined printers.

  This exit is invoked for all data-related functions in NPF processing.

  The name of this exit is user-dependent. It must follow MVS module naming standards, can be up to an 8-byte alphanumeric name and can be defined in the:

  – RTDAIREX field of the routing data area.

  – IREXIT parameter in "EZAPPFL TYPE=OPTIONS" on page 50.

  – INPUT RECORD EXIT field, as illustrated in Figure 18 on page 72.

  See "Using the EZAPPFCD macro" on page 108 to produce the DSECTS needed for the exits.

Writing the general routing exit

The general routing exit is used when the installation can provide a more sophisticated and accurate method of routing than the Network Print Facility can with the routing and options files. The exit can interpret information provided by the capture points in an installation-specific context and produce more accurate routings. For example, examination of the JES or VTAM specific data can produce a direct routing or a different way to access the routing file to produce a routing.

The general routing exit can be used for two purposes:

- Create a routing data area using an algorithm different from that provided by the Network Print Facility. In this use of the general routing exit, the exit interprets the input parameters and develops the information required for the routing data area. After the routing data area is built, the exit returns control to the Network Print Facility specifying that the routing data area has been built. Optionally, the exit can cause the invocation of a specific routing exit by putting the name of the specific routing exit in the RTDASREX field in the routing data area. Otherwise, this field should be set to either binary zeros or blanks.

- Change the key used to access the routing file. In this use of the general routing exit, the exit interprets the input parameters and develops the major and minor keys for access to the routing file. The exit returns these keys to the Network Print Facility and a code indicating that the keys have been changed. The system then uses the keys to access the routing (and then options) file and builds the routing data area based on the information in the files. The use of the specific routing exit is controlled by an entry in the routing file (see the SREXIT parameter in the ROUTING function of the EZAPPFL macro.

Environment

The exit should be coded following standard assembly linkage with the following parameters:

Name
EZAPPGPR

AMODE
Any

RMODE
24 or 31

Link Attribute
RENT (Reentrant)

In the JES environment, it runs in supervisor state using storage key 1. In the VTAM environment, it runs in problem state.

Processing

At initialization, the Network Print Facility issues a conditional LOAD for the module EZAPPGPR. If the LOAD fails, the Network Print Facility assumes that the general routing exit is not supplied and builds the routing data area for each print file using the routing and options files. If the LOAD is successful, the Network Print Facility performs the following initialization steps for each print file.

- Invoke EZAPPGPR
  - If EZAPPGPR built a routing data area:
    - Invoke a specific routing exit if one was specified
  - If EZAPPGPR replaced the keys or did nothing:
    - Build a routing data area using the routing and options files.
    - Invoke a specific routing exit if one was specified

Note: To maintain integrity of the routing and options files, do not use the general routing exit to process these files.

Installing the exit

The exit should be linked into a target library which is referred to by a JOBLIB, STEPLIB, or LINKLIST entry for execution of the appropriate capture point.

Input

See “Parameter list contents” on page 96.

Registers at entry

GPR1
Parameter list

GPR13
Caller's save area

GPR14
Caller's return address

GPR15
Entry point of exit

Parameter list contents

The macro EZAPPFCD GRE=DSECT generates a DSECT of the parameter list for the general routing exit. EZAPPFCD GRE=INLINE generates an inline copy.

See "Using the EZAPPFCD macro" on page 108 to generate the DSECTS you need.

GRCALLID
Address of a four-byte character field with the identity of the caller:

Value
Meaning

'JES '
The caller is a JES external writer.

'VTAM'
The caller is the VTAM application module.

GRCALLPM
The address of the JES or VTAM-specific parameter area. For JES, it is the address of a list of parameters that consists of:

1. A pointer to the GETDS function-dependent area in the IAZFSIP DSECT. (To generate this DSECT, code the IAZFSIP macro with LIST=YES in your exit.)
2. A pointer to the values of the PARM parameters passed by the NPF FSS writer start procedure.
3. A fullword binary number specifying the length of the PARM parameters passed by the NPF FSS writer start procedure.
4. A fullword of binary zeros
5. A fullword of binary zeros (reserved)
6. A fullword of binary zeros (reserved)
7. A fullword of binary zeros (reserved)
8. The EOF flag

For VTAM, it consists of the address of the 35-byte BIND image received from VTAM.

GRRTNAME
Address of an area containing two 8-byte name fields. The names passed to the exit are the major and minor names of the routing record that NPF will use unless otherwise directed. To use the NPF-selected routing record, return with GPR 15 set to 0000. To specify a different routing record, return with GPR 15 set to 0001 and supply the new major and minor names in this area.

GRRTDATA
The address of the fullword where the exit will store the address of the new routing data area. This field must be supplied by the exit when the return code is 0002.

Because NPF will FREEMAIN the new routing area allocated by the EZAPPGPR exit, make sure the size of the RDA corresponds to the following formula:

```
length=346+(RTDADEST*516)
```

Return codes for the general routing exit

Upon completion the general routing exit should return a value in GPR 15 as follows:

0000
Exit completed successfully. Will use the routing record specified by the major and minor names that were passed to the exit.

0001
Exit completed successfully. Will use the routing record specified by the new major and minor names that were supplied by the exit.

0002
Exit completed successfully. No routing record will be used because the routing data area was supplied by the exit.

Anything Else
Exit did not complete successfully. Message EZY0631E is accompanied by either:

- For VTAM - sense code x'081C0103'
- For JES - return code 1002

Output

See "Parameter list contents" on page 96

Registers at exit

GPR15
As defined above

All other registers
Restored to caller's values

Writing the specific routing exit

The specific routing exit is used to modify the contents of the routing data area. It is used much the same as the general routing exit. NPF invokes it after it builds the routing data area, but unlike the general routing exit, the user can specify a specific exit for each routing. Like the general routing exit, the specific routing exit can use information provided by the capture points in an installation-specific context and produce more accurate routings. For example, examination of the JES or VTAM specific data might produce a more accurate routing than is contained in the routing file or produced by the general routing exit.

The specific routing exit is invoked after the routing data area (RDA) has been built. Because the routing data area can be built by either the general routing exit or by data from the routing and options files, the specific routing exit can be used with either method. Its purpose is to further modify the routing data area.

See Appendix B, "Network Print Facility's Routing Data Area (RDA)," on page 135 for the structure of the routing data area. When specifying routing data format, be careful to leave the specific routing exit field (RTDAEXIT field in the RDA, the SREXIT parameter in the ROUTING function of the EZAPPFL macro, and the ROUTING EXIT field, as illustrated in Figure 27 on page 79) blank if you are not using a specific routing exit. The required fields are the same as in the general routing exit. The run-time environment is the same as the general routing exit.

Environment

The exit should be coded following standard assembly linkage with the following parameters:

Name
User Specified. The name must follow MVS conventions for Load module names and must be specified in the routing file or placed in the routing data area by the general routing exit.

AMODE
Any

RMODE
24 or 31

Link Attribute
RENT (Reentrant)

Processing

The system tests for the presence of the specific routing exit by examining field RTDASREX in the routing data area. If this field is zeros or blanks, the Network Print Facility assumes no specific routing exit is to be used. If this field is something else, it uses this name in a conditional LINK to the specific routing exit.

Installing the exit

The exit should be linked into a target library which is referred to by a JOBLIB, STEPLIB, or LINKLIST entry for execution of the appropriate capture point.

Input

See "Parameter list contents" on page 98

Registers at entry

GPR1
Parameter list

GPR13
Caller's save area

GPR14
Caller's return address

GPR15
Entry point of exit

Parameter list contents

The macro EZAPPFCD SRE=DSECT generates a DSECT of the parameter list for the specific routing exit. EZAPPFCD SRE=INLINE generates an inline copy.

SRCALLID
Address of a four-byte character field with the identity of the caller:

Value	Meaning
'JES '	The caller is a JES external writer.
'VTAM'	The caller is the VTAM application module.

SRCALLPM
The address of the JES or VTAM-specific parameter area. For JES, it is the address of a list of parameters that consists of:

1. A pointer to the GETDS function-dependent area in the IAZFSIP DSECT. (To generate this DSECT, code the IAZFSIP macro with LIST=YES in your exit.)
2. A pointer to the values of the PARM parameters passed by the NPF FSS writer start procedure.
3. A fullword binary number specifying the length of the PARM parameters passed by the NPF FSS writer start procedure.

4. A fullword of binary zeros
5. A fullword of binary zeros (reserved)
6. A fullword of binary zeros (reserved)
7. A fullword of binary zeros (reserved)
8. The EOF flag

For VTAM, it consists of the address of the 35-byte BIND image received from VTAM.

SRRTDATA
A fullword binary field specifying the address of the routing data area. The exit modify this area if desired.

Return codes for the specific routing exit

Upon completion the exit will return a value in GPR 15 as follows:

0000
Exit completed successfully.

Anything Else
Exit did not complete successfully. Message EZY0632E is accompanied by either:

- For VTAM - sense code x'081C0103'
- For JES - return code 1002

Or, in the case of a failure to link, message EZY0636E is accompanied by either:

- For VTAM - sense code x'081C0103'
- For JES - return code 1005

Output

See "Parameter list contents" on page 98

Registers at exit

GPR15
As defined above

All other registers
Restored to caller's values

Writing the input record exit

The input record exit can be used for any editing function that you want. It is invoked for all data-related functions in NPF processing, and can play an integral part in the creation of the file being transmitted. In addition, it can modify routing data based on print data observed, for example, the user's name or system name.

The input record exit allows the system administrator to modify the contents of printed output.

There are two purposes for the input record exit

- To modify, add to, or remove print data from the print stream.
- To modify the routing data area based on print data.

Understanding how LPR has been defined to operate in a particular environment will enable installations to take advantage of the input record exit's capabilities. The input record exit can:

- Add data, such as a banner page.

- Modify data. The input record exit is invoked with each PUT. The input record exit can modify data (for example, by translation or encryption) and then continue with the PUT. The input record exit has storage that the Network Print Facility can manage for you after you construct it.
- Delete data.
- Cause a file to be rescanned (only for JES output).

Environment

The exit should be coded following standard assembly linkage with the following parameters:

Name
User Specified. The name must follow MVS conventions for Load module names and must be specified in the Options file or placed in the routing data area by the general routing exit.

AMODE
Any

RMODE
24 or 31

Link Attribute
RENT (Reentrant)

Processing

The system tests for the presence of the input record exit by examining field RTDAIREX in the routing data area. If this field is zeros or blanks, the Network Print Facility assumes no input record exit is to be used. If this field is something else, it uses this name in a conditional LOAD to the input record exit and saves its address. This address is used to access the exit on all calls.

The input record exit is invoked for 4 conditions. The specific condition can be determined by checking the field RTDAFUNC in the routing data area. The four conditions explained below are: open, put, close, and term/release.

OPEN
The system is preparing to process a print file. It is about to allocate and open the QSAM file to contain the data to be passed. The input record exit should prepare itself as follows:

1. Obtain program storage, if necessary. For the JES capture point, this should be necessary only on the first OPEN issued by a (JES) logical printer. For the VTAM capture point, this is necessary for each print file. The address of this storage is stored in the area pointed to by IRESTRG and is returned for any processing related to this logical printer/file.
2. Get addressability to the routing data area. The address of the RDA is in field IRERTD in the parameter area. Use the DSECT RTDATA, generated using the EZAPPFCD RDA=DSECT macro.
3. Examine the field RTDARTYP. If it is not acceptable, it can be changed to one of the following record fowmats:

   Value
   Meaning

   FB
   File will be allocated as fixed blocked records.

   FBA
   File will be allocated as fixed blocked records with ASA carriage control.

   FBM
   File will be allocated as fixed blocked records with machine carriage control.

   VB
   File will be allocated as variable blocked records.

   VBA
   File will be allocated as variable blocked records with ASA carriage control.

VBM
File will be allocated as variable blocked records with machine carriage control.

4. Examine the field RTDARECL. This field specifies the logical record length to be used for the QSAM file. If the file format is VB, VBA or VBM, this specifies the maximum record length. If the file format is FB, FBA or FBM, it specifies the actual record length. If the value is not acceptable, it can be changed but with the following considerations:

   - If the record format is FB, FBA or FBM, and this length is greater than the length of records passed from the capture point, the records are padded to the right with blanks. This might cause unpredictable results including the insertion of additional blank lines on the printed output. The exit must avoid this condition by reformatting the data as it is processed.
   - If the record format is FB, FBA or FBM, and this length is less than the length of records passed from the capture point, the records are split, resulting in multiple printed lines for each line received from the capture point. The exit must avoid this condition by reformatting the data as it is processed.
   - If the record format is VB, VBA or VBM, this length is the maximum record length and includes the 4-byte LLbb field preceding each record.

5. Examine the field RTDABLKS. This field specifies the blocksize to be used in the QSAM file. If the value is not acceptable it can be changed, with the following considerations:

   - If the record format is FB, FBA or FBM, it must be an even multiple of the value in RTDARECL.
   - If the record format is VB, VBA or VBM, it must be a minimum of the value in RTDARECL plus 4.

6. Examine the field RTDABUFS. This field specifies the buffer size to be passed to the exit for PUT processing. If this buffer size is not acceptable, it can be changed. It must have a minimum value of the value in RTDARECL plus 2.

PUT
Network Print Facility is about to place a record of print data on the file for processing. The data is located in a buffer and the address and length of the data is passed to the input record exit via the parameter list. The input record exit performs three functions:

1. It modifies the data. For this purpose, there are two bytes of buffer space in front of the data, which is used by the exit to expand the record to the left. The amount of buffer space to the right is equal to the buffer size minus the record length minus 2. If the exit moves the record within the buffer or moves it to another buffer, it must replace the address and length of the data in the parameter list fields.

2. The input record exit specifies the disposition of the data by using the IREWFLG as follows:

   Value
   Meaning

   X'00'
   Write the data to the file.

   X'01'
   Do not write the data to the file.

3. The input record exit controls the logic of the capture point using the field IRERFLG as follows:

   Value
   Meaning

   X'00'
   Do not re-invoke the input record exit until the next data record has been obtained or the file is to be closed.

   X'01'
   Re-invoke the input record exit without a new input buffer before returning to the capture point. This permits the input record exit to insert additional data.

X'02'

This option is available for JES only. It tells the capture point to restart processing at the beginning of the spool file.

CLOSE

Processing for CLOSE is similar to processing for PUT. The following exceptions must be observed:

- The data address and length fields will have no meaning at invocation. If the exit wants to write an additional data record to the file, it must place the address and length of the data in these fields.
- Unless the input record exit provides data to be written to the file, IREWFLG must be set to X'01'.
- A value of X'00' in IRERFLG is interpreted as 'continue with CLOSE processing. Thus, if the exit specifies this value, it will be the last time the exit is invoked related to this file.
- A value of X'02' in IRERFLG is currently not supported for CLOSE.

TERM/RELEASE

The exit is invoked at this time to allow it to free any resources such as storage that it might have obtained.

Special considerations for handling JES data

The following information should be taken into consideration when coding an input record exit for the JES capture point:

- The default record format for all QSAM data sets is RECFM=VBA if SPIN=GROUP is specified. For SPIN=DATASET processing, the RECFM will be VBA, VBM, or VB depending on the type of carriage control specified when the data set was written to JES (ASA=VBA, machine=VBM, none=VB).
- The default DCB information for the QSAM file will be LRECL=4092 and BLKSIZE=4096. If a larger LRECL is needed, the input record exit must override the LRECL and BLKSIZE. Record lengths larger than 4092 are not supported unless the input record exit is coded.
- At OPEN time, any of the characteristics (RECFM, LRECL, BLKSIZE) of the QSAM data set can be overridden by the input record exit. However, it is the responsibility of the input record exit code to ensure the data records conform to any changes.

Installing the exit

The exit should be linked into a target library which is referred to by a JOBLIB, STEPLIB, or LINKLIST entry for execution of the appropriate capture point.

Input

See "Parameter list contents" on page 102

Registers at entry

GPR1

Address of the parameter list

GPR13

Caller's save area

GPR14

Caller's return address

GPR15

Entry point of exit

Parameter list contents

The options file can specify an input record exit to be invoked before putting a print record in the print file. This is the parameter list passed to that input record exit. The name of the exit is specified by the user in the IREXIT parameter on the OPTIONS function of the EZAPPFL macro (see "EZAPPFL TYPE=OPTIONS" on page 50).

The parameter list consists of the address of a pointer to the following parameter area. The macro EZAPPFCD IRE=DSECT generates a DSECT of this parameter area. EZAPPFCD IRE=INLINE generates an inline copy.

IRELLEN
Length of this parameter area

IRESTRG
The address of a working storage area used by the exit. The length of this area is 16 bytes.

IRECPAD
The address of the JES or VTAM-specific parameter area. For JES, it is the address of a list of parameters that consists of:

1. A pointer to the GETDS function-dependent area in the IAZFSIP DSECT. (To generate this DSECT, code the IAZFSIP macro with LIST=YES in your exit.)
2. A pointer to the values of the PARM parameters passed by the NPF FSS writer start procedure.
3. A fullword binary number specifying the length of the PARM parameters passed by the NPF FSS writer start procedure.
4. A pointer to the IDXFLAG1 byte in the IAZIDX DSECT. (To generate this DSECT, code the IAZIDX macro with LIST=YES in your exit.)
5. A fullword of binary zeros (reserved)
6. A fullword of binary zeros (reserved)
7. A fullword of binary zeros (reserved)
8. The EOF flag

For VTAM, it consists of the address of the 35-byte BIND image received from VTAM.

IREBPTR
The address of the input/output buffer

IREDLEN
A fullword binary number specifying the length of the data in the buffer.

IREWFLG
A one-byte field specifying whether the data in the buffer is to be written to the file or skipped. On entry to the exit, this field is set to X'00'. On return from the exit, it is set as follows:

X'00'
The data is to be written to the file.

X'01'
The data is not to be written.

IRERFLG
A one-character field which specifies what input record is to be returned on the next call to the exit. On entry to the exit, this flag is set to X'00'. On return from the exit, it is set as follows:

X'00'
Specifies the next record returned to the exit will be the next record from the input spool file.

X'01'
Specifies that control will be returned to the exit so that additional records can be inserted into the output data stream. If the exit returns a new I/O buffer address BELOW the 16 MB line in IREBPTR, then subsequent invocations of the exit will be passed the address of the CURRENT input record in IREBPTR. If the exit returns a new I/O buffer address ABOVE the 16 MB line, or IREBPTR is not changed, then subsequent invocations of the exit will be passed the address of the last output buffer in IREBPTR.

X'02'
Specifies that the next record returned to the exit will be the first record in the spool file.

IREEOFF

A one-character field used to indicate that an EOF has been encountered.

Value

Meaning

x'00'

Not EOF

x'01'

EOF

IRECID

A one-character field used to identify the caller.

Value

Meaning

V

VTAM

J

JES

IREOPLN

The length of the data set name plus the options data already used by NPF. Subtract this from 255 to determine how much space you have for user-supplied options.

IREABNF

A one-character field which indicates if the input record exit can insert records into the output data stream. On entry to the exit, this flag is set to:

X'00'

Specifies that print data set is open and the exit can insert records into the file.

X'02'

Specifies that print data set is closed due to an out of space condition and no further records can be inserted into the output file.

IRERTD

The address of the routing data area.

```
*----------------------------------------------------------------------*
*                INPUT RECORD EXIT PARAMETER LIST                      *
*----------------------------------------------------------------------*
IRELIST  DS    0F
IRELLEN  DS    F                  Length of this parameter list
IRESTRG  DS    F                  Pointer to static 16-byte memory
IRECPAD  DS    F                  Pointer to JES/VTAM parameter list
IREBPTR  DS    F                  Pointer to 32K I/O buffer
IREDLEN  DS    F                  Length of data in the buffer
IREWFLG  DS    C                  Write/Skip flag
IRERFLG  DS    C                  Read flag
IREEOFF  DS    C                  EOF encountered flag
IRECID   DS    C                  Caller's ID (J=JES V=VTAM)
IREOPLN  DS    H                  Length of the DSN  + options data
IREABNF  DS    C                  Print file abend indicator
         DS    C                  Reserved
IRERTD   DS    F                  Address of the routing data area
IRELISTL EQU   *-IRELIST          Length of Parameter List
```

Figure 40. Input exit record parameter list

Return codes for the input record exit

Upon completion, the exit will return a value in GPR 15 as follows:

0000

Exit completed successfully.

Anything Else

Exit did not complete successfully. Message EZY0660E is accompanied by either:

- For VTAM - sense code x'081C0109'
- For JES - return code 3003

Output

See “Parameter list contents” on page 102

Registers at exit

GPR15
As defined above

All other registers
Restored to caller's values

SPIN=GROUP restrictions

NPF provides support in the JES capture point for SPIN=GROUP processing when the data sets in the group have different characteristics. To correctly process data sets with different characteristics, the following restrictions apply when SPIN=GROUP is specified:

- If a record format for the QSAM data sets other than RECFM=VBA is required, then the input record exit must be coded to override the RECFM.
- If the RECFM is overridden by the input record exit and any record contains carriage control, NPF will NOT strip off the carriage control. It will be passed as the first character of the data block.
- The IDX flags are checked for every SYSOUT record. If neither the ANSI nor the machine carriage control characters are set, a DEFAULT new line Carriage Control is ADDED in the first byte. If the input record exit must check if this carriage control byte has been added, the address of the IDXFLAG1 byte is passed in the JES-specific parameter area (IRECPAD). If IDXANSI (x'20') and IDXMACH (x'10') are zero, the carriage control byte has been added.
- All output created by SPIN=GROUP processing will have carriage control unless overridden by the input record exit. Avoid coding the NOCC LPR option in the options file unless an input record exit will be used to override the VBA format.
- For SPIN=GROUP processing, the parameter list passed to the input record exit contains a pointer to the GETDS parameter from the currently ACTIVE data set, not the FIRST data set in the group.
- The JES JOB Separator Page Data Area (JSPA) is addressable only from the GDSJSPA field of the current GETDS parameter. For SPIN=GROUP processing, the JSPA is for the CURRENT data set in the group only. If information is needed from the JSPA of the first data set while processing subsequent data sets in the group, the input record exit must make its own copy of the required data.

Input record exit to delete leading blank pages

The following information shows an example of an input record exit that deletes leading blank pages from the printed output. You can copy this sample from *hlq*.SEZAINST(DELBLANK).

```
DELBLANK TITLE 'Input Record Exit to Delete Leading Blank Pages'
*----------------------------------------------------------------------*
*   DELBLANK NPF Sample Input Record Exit                              *
*----------------------------------------------------------------------*
*                                                                      *
*   COPYRIGHT = NONE                                                   *
*                                                                      *
*   SMP/E Distribution Name: EZAEC043                                  *
*                                                                      *
*   Some applications generating SNA print data streams force a new    *
*   page at the beginning of independent pieces of work. This       *
*   correspond with the NPF end-of-file rule specified.                *
*                                                                      *
*   Some LPD print managers force a new page between files.            *
*                                                                      *
*   If both are true, the printed output will begin with a blank       *
*   page. This NPF input record exit looks at the carriage control     *
*   character on the first line of output in files from the NPF VTAM  *
*   capture point application. If it is a form feed, it is changed    *
*   to an overstrike so that printing will begin on the first page.   *
*                                                                     *
*---------------------------------------------------------------------*
DELBLANK CSECT
DELBLANK AMODE ANY
DELBLANK RMODE ANY
*              Registers           entry       DELBLANK    exit
R0       EQU   0                   n/a         work        =entry
R1       EQU   1                   >>IRELIST   work        =entry
R2       EQU   2                   n/a         work        =entry
R3       EQU   3                   n/a         work        =entry
R4       EQU   4                   n/a         work        =entry
R5       EQU   5                   n/a         work        =entry
R6       EQU   6                   n/a         work        =entry
R7       EQU   7                   n/a         work        =entry
R8       EQU   8                   n/a         >IRXSTATD   =entry
R9       EQU   9                   n/a         >RTDATA     =entry
R10      EQU   10                  n/a         >IRELIST    =entry
R11      EQU   11                  n/a         base 2      =entry
R12      EQU   12                  n/a         base 1      =entry
R13      EQU   13                  >callsave   >DELBSTG    =entry
R14      EQU   14                  >return     work        =entry
R15      EQU   15                  >DELBLANK   retcode     retcode
         SPACE 5
         USING DELBLANK,R15       Establish temporary base register
         B     DELB0000           Branch around constants
         DC    CL9'DELBLANK'      Module identifier
         DC    CL9'&SYSDATE'      Assembly date
         DC    CL6'&SYSTIME'      Assembly time
DELBASE2 DC    A(DELBLANK+4096)   Second base register if needed
         SPACE 5
DELB0000 DS    0H
         STM   R14,R12,12(R13)    Save caller's registers
         LR    R12,R15            Change base registers
         DROP  R15                Tell assembler
         USING DELBLANK,R12,R11
         L     R11,DELBASE2       Establish second base register
         L     R10,0(R1)          Get address of exit parameter list
         USING IRELIST,R10        Make exit parameter list addressable
         L     R9,IRERTD          Get address of routing data area
         USING RTDATA,R9          Make routing data area addressable
         L     R8,IRESTRG         Get address of IRE static memory
         USING IRXSTATD,R8        Make IRE static storage addressable
         ICM   R1,15,IRXDSTGA     Get address of IRE dynamic storage
         BNZ   DELB0020           Storage has already been allocated
         LA    R2,DELBSTGL        Get length of program storage
         GETMAIN RC,LV=(R2),LOC=ANY Get program storage
         LTR   R15,R15            Did GETMAIN work?
         BZ    DELB0010           Yes
         WTO   'DELBLANK GETMAIN FAILED'
         LA    R15,8              No,  show error
         B     RET00010           Return to NPF
DELB0010 DS    0H
         ST    R1,IRXDSTGA        Save dynamic storage address
DELB0020 DS    0H
         ST    R1,8(0,R13)        Complete save area pointers
         ST    R13,4(0,R1)
         LR    R13,R1             Point to dynamic storage area
         USING DELBSTG,R13        Tell assembler
         SPACE 5
         CLC   RTFUNCTN(4),=CL8'OPEN' Is this OPEN call?
         BE    OPN00000           Yes
         CLC   RTFUNCTN(3),=CL8'PUT'  Is this PUT call?
         BE    PUT00000           Yes
         CLC   RTFUNCTN(5),=CL8'CLOSE' Is this CLOSE call?
         BE    CLS00000           Yes
         CLC   RTFUNCTN(7),=CL8'RELEASE' Is this RELEASE call?
         BE    REL00000           Yes
         CLC   RTFUNCTN(4),=CL8'TERM' Is this TERM call?
         BE    TRM00000           Yes
         WTO   'DELBLANK UNKNOWN FUNCTION'
         LA    R15,4              Bad function, show error
         B     RET00000           Return to NPF
         SPACE 5
*---------------------------------------------------------------------*
*        Process OPEN call
*---------------------------------------------------------------------*
OPN00000 DS    0H
         MVI   IRXFLAG,IRXNEW     Set flag for new file
         XR    R15,R15            Zero return code
         B     RET00000           Go return to NPF
*----------------------------------------------------------------------*
*        Process PUT call
*----------------------------------------------------------------------*
PUT00000 DS    0H
         CLI   IRXFLAG,IRXNEW     First PUT to this file?
         BNE   PUT00020           No, pass it through
         MVI   IRXFLAG,IRXOLD     Only look at first PUT
         CLI   IRECID,C'V'        Is this VTAM application?
         BNE   PUT00020           No, pass it through
         L     R2,IREBPTR         Get input buffer pointer
         CLI   RTDARTYP,C'V'      Variable length records?
         BNE   PUT00010           No, continue
         LA    R2,4(0,R2)         Yes, point past record descriptor
PUT00010 DS    0H                 R2 points to carriage control
         CLI   0(R2),C'1'         Is it form feed?
         BNE   PUT00020           No, pass it through
         MVI   0(R2),C'+'         Change to overstrike
PUT00020 DS    0H
         MVI   IREWFLG,X'00'      Write this record
         MVI   IRERFLG,X'00'      Read next record
         XR    R15,R15            Zero return code
         B     RET00000           Go return to NPF
*----------------------------------------------------------------------*
*        Process CLOSE call
*----------------------------------------------------------------------*
CLS00000 DS    0H
         MVI   IREWFLG,X'01'      No record to write
         MVI   IRERFLG,X'00'      Normal return from CLOSE
         XR    R15,R15            Zero return code
         B     RET00000
*----------------------------------------------------------------------*
*        Process RELEASE or TERM call
*----------------------------------------------------------------------*
REL00000 DS    0H
TRM00000 DS    0H
         LA    R2,DELBSTGL        Get length of dynamic storage
         LR    R1,R13             Get address of dynamic storage
         L     R13,SAVEAREA+4     Get address of caller's save area
         FREEMAIN RU,LV=(R2),A=(R1) Free dynamic storage
         XR    R15,R15            Zero return code
         B     RET00010
         SPACE 5
*----------------------------------------------------------------------*
*        Return to NPF
*----------------------------------------------------------------------*
RET00000 DS    0H                 Switch back to caller's save area
         L     R13,SAVEAREA+4     Return to caller
RET00010 DS    0H                 Already back to caller's save area
         ST    R15,16(R13)        Set return code in saved R15
         LM    R14,R12,12(R13)    Restore caller's registers
         BSM   0,R14
         EJECT
*----------------------------------------------------------------------*
*        Program constants
*----------------------------------------------------------------------*
         LTORG
         SPACE 5
IRXSTATD DSECT                    16 byte static IRE storage
IRXDSTGA DS    A                  Address of dynamic storage area
IRXFLAG  DS    X                  File status flag
IRXNEW   EQU   X'00'              ... first record in file
IRXOLD   EQU   X'FF'              ... not first record in file
         DS    3X                 unused
         DS    D                  unused
         SPACE 5
DELBSTG  DSECT
SAVEAREA DS    18F                Save Area
         DS    0D                 Round up to double word boundary
DELBSTGL EQU   *-DELBSTG          Length of dynamic storage area
         SPACE 5
         EZAPPFCD IRE=DSECT,   DSECT for input record exit parameters  X
               RDA=DSECT       DSECT for routing data area
         SPACE 5
         END   ,                   End of DELBLANK  module
```

The Routing Data Area (RDA)

This data area (for a layout see Appendix B, “Network Print Facility's Routing Data Area (RDA),” on page 135) is the cornerstone of the Network Print Facility. It contains the routing information referenced by the Network Print Facility. The Network Print Facility can build the RDA, or the user can build it. In either case, it must be established so the Network Print Facility can access the routing information required.

The routing data area is accessible to all three exits. If the general routing exit builds the routing data area, it should initialize all fields. Fields that can be modified by either the specific routing exit or the input record exit are marked.

Use the general routing exit (EZAGRPPR) to create the routing data area.

Both the general routing exit and the specific routing exit affect the RDA. The Network Print Facility modifies the RDA depending on the result of a series of checks, as follows:

1. Does a general routing exit exist?

   Invoke it. The general routing exit does one of the following thing:

   a. Builds the RDA (don't read the routing and options files).
   b. Modifies the keys used to read the routing and options files.
   c. Nothing.
2. The GRE notifies NPF of the results of this check. If no RDA was built, NPF builds one.

   The system then checks if a specific routing exit exists. If there is one, the Network Print Facility invokes it.

   The specific routing exit can modify the RDA.

See Appendix B, “Network Print Facility's Routing Data Area (RDA),” on page 135 for the exact structure of the RDA.

Using the EZAPPFCD macro

Use the EZAPPFCD macro to generate the data areas you need and to supply the appropriate information for the RDA. This macro provides copies of parameter lists and record formats for the Network Print Facility. All specifications are optional.

Environment
MVS Assembler

Programming Requirements
The use of YES/DSECT options creates a DSECT. Your program need to resume the previous CSECT. Use of the INLINE option creates parameter lists or record descriptions within the current CSECT/ DSECT.

Restrictions
None

Input Regs
Not Applicable

Output Regs
Not Applicable

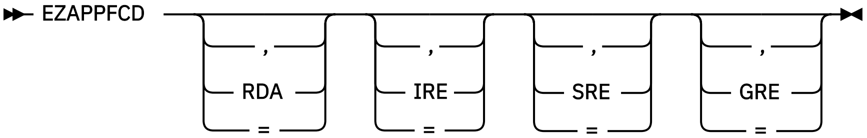
Performance Implications
None

Abend Codes
None

Return Codes
None

Syntax

Macro Syntax:



Parameters

RDA=YES|DSECT|NO

YES, DSECT
Generate Routing Data Area DSECT

NO
Do not generate Routing Data Area DSECT

IRE=YES|DSECT|NO|INLINE

YES, DSECT
Generate Input Record Exit Parameter List DSECT

NO
Do not generate Input Record Exit Parameter List

INLINE
Generate Input Record Exit Parameter List definition

SRE=YES|DSECT|NO|INLINE

YES, DSECT
Generate Specific Routing Exit Parameter List DSECT

NO
Do not generate Specific Routing Exit Parameter List

INLINE
Generate Specific Routing Exit Parameter List definition

GRE=YES|DSECT|NO|INLINE

YES, DSECT
Generate General Routing Exit Parameter List DSECT

NO
Do not generate the General Routing Exit Parameter List

INLINE
Generate General Routing Exit Parameter List definition

Example

```
EZAPPFCD ,                                                  X
      IRE=DSECT          Generate DSECT for Input Record Exit
```

Chapter 8. Operating the Network Print Facility

To start the Network Print Facility, you need separate JCL procedures for each instance of the NPF FSS writer, the VTAM capture point, and the queue manager. Similarly, to terminate the Network Print Facility or modify it while it is running, you need to perform separate actions for each part of the Network Print Facility.

The FSS writers are started by JES console commands. Both the VTAM capture point and queue manager can be submitted as batch jobs or converted to procedures, stored in a system PROCLIB, and started from the console with the MVS START command.

Note: z/OS Communications Server must be running before you start the NPF VTAM capture point, FSS writer, or queue manager programs.

The chapter includes information on how to:

- Start, stop, and operate the NPF FSS Writer
- Start, stop, and operate the VTAM Capture Point
- Start, stop, and operate the Queue Manager

TCP/IP High-Level Qualifier

The high-level qualifier is determined as part of the overall TCP/IP configuration and is explained in “TCPIP.DATA configuration” on page 16. It can be specified in TCPIP.DATA with the DATASETPREFIX statement, or left to the system default value. If you are using the //SYSTCPD DD statement to specify TCPIP.DATA, ensure that ISPF can find your defined high-level qualifier by including this statement in:

- All NPF (JES and VTAM) capture point startup procedures
- All queue manager start up procedures
- TSO/E ISPF user logon procedure

Operating the NPF FSS writer

Before you can start and operate an FSS writer you need to have:

- The FSS defined to JES (FSSDEF)
- A start procedure catalogued in the system or other recognized PROCLIB (the catalogued procedure must have access to the JES program named HASPFSSM).

After these have been defined to your system, you can enter commands at the operator console to start and stop the individual FSS writers which are running as FSA subtasks.

On some systems, you can also stop the entire FSS address space with the CANCEL command. For example, to stop the address space of a functional subsystem defined as FSSNAME=TCPFSS or FSSDEF(TCPFSS), you would enter:

```
C TCPFSS
```

Creating the NPF FSS writer start procedure

The following information shows a sample procedure to start an NPF FSS writer. You can copy this sample from *hlq*.SEZAINST(FSWTR), modify it to suit your environment, and catalogue it to a recognized PROCLIB, such as the SYS1.PROCLIB.

Note: The log file cannot be shared. If you want to run more than one instance of the NPF FSS writer, you will need separate procedures with a unique name for this data set in each procedure.

```
//FSWTR  PROC
//* ------------------------------------------------------------------*
//*
//* COPYRIGHT = NONE
//*
//*  SMP/E Distribution Name: EZAEC0YU                                 *
//*
//* This JCL starts the Network Print Facility's JES interface.
//*
//*   The executable code is assumed to be in a link library which is
//*   in the system link list.  (The default is TCPIP.SEZALINK).
//*
//*   REGION=3M is specified to provide sufficient below-the-line
//*   storage for a writer with a single FSA (printer) subtask.
//*   That number should be increased by 850K for each additional
//*   FSA (printer) subtask defined for the writer.
//*
//*   In this sample, the names of the routing, options and queue
//*   files are:
//*
//*       TCPIP.ROUTING
//*       TCPIP.OPTIONS
//*       TCPIP.QUEUE
//*
//*   All three are assumed to have been created and initialized
//*   prior to the use of this JCL.
//*
//*   The log data set for this sample is TCPIP.JES1.LOG.
//*
//*   NOTE: All temporary data sets created by Network Print Facility
//*         will have the high-level qualifier specified via the
//*         defined search sequence.  (See "Configuring NPF Using
//*         TCPIP.DATA and NPF.DATA Statements" in IBM TCPIP for MVS:
//*         Network Print Facility).
//*
//*         In this sample, user specified defaults will be found
//*         in the TCPIP.DATA data set defined by the SYSTCPD DD
//*         statement and in the NPF.DATA data set defined by the
//*         SYSNPFD DD statement.
//*
//* ------------------------------------------------------------------*
//IEFPROC  EXEC  PGM=EZAPPFS,REGION=3M
//EZAPPRF  DD DSN=TCPIP.ROUTING,DISP=SHR
//EZAPPOF  DD DSN=TCPIP.OPTIONS,DISP=SHR
//EZAPPQF  DD DSN=TCPIP.QUEUE,DISP=SHR
//EZAPPLOG DD DSN=TCPIP.JES1.LOG,DISP=OLD
//SYSTCPD  DD DSN=TCPIP.TCPIP.DATA,DISP=SHR
//SYSNPFD  DD DSN=TCPIP.NPF.DATA,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSMDUMP DD SYSOUT=*
```

Modifying the NPF FSS writer start procedure

ESTAE-SDUMP, SPIN, and MSG-SUPPRESS are a few of the run-time options available for the NPF FSS writer.

ESTAE-SDUMP=YES|NO

This option specifies whether the ESTAE routine will issue an SDUMP (SVC dump) whenever an FSS or FSA task abends. Specification of this parameter immediately affects all subsequent task abends.

YES

Specifies an SDUMP will be issued. This is the default.

NO

Specifies an SDUMP will not be issued.

SPIN=DATASET|GROUP

This specifies how the data sets received from JES should be written to output data sets.

DATASET

Specifies an output data set will be written for each data set received from JES.

GROUP

Specifies an output data set will be written for each group of data sets received from JES. This option is appropriate for JES2 users and is the default.

A data set group starts when JES sends a START or CONTINUE job separator page area (JSPA) and ends when JES sends an END JSPA or another START or CONTINUE JSPA.

Specification of this parameter affects only the SYSOUT data sets that have not yet been selected by the FSA. Data set that have already been selected use the previous setting.

In JES3 environments, use the SPIN=DATASET option on the NPF FSS writer to separate the output into individual data sets.

QSAMLRECL=4092 32756

This option specifies the logical record length (LRECL) to be used when allocating the print data set.

4092

Specifies an output data set LRECL of 4092, which is the default. Specifying 4092 instructs NPF to process output as if the maximum LRECL is 4088 bytes or less. If you specify 4092 and the output records are greater than 4088 bytes, the FSS writer will ABEND.

32757

Specifies an output data set LRECL of 3275. Specifying 32756 allows NPF to process output with LRECLs up to 32752 bytes. You must specify 32756 if any output records are larger than 4088 bytes.

MSG-SUPPRESS=NONE

This debugging option should be used only at the direction of the IBM Support Center to gather additional trace data for problem resolution.

You can establish or modify the run-time options for the NPF FSS writer procedure in three ways.

- With the PARM field on the EXEC statement in the NPF FSS writer start procedure
- With a PARMLIB DD statement in the NPF FSS writer start procedure
- Dynamically, using the MVS modify command

The system processes options in the above order. If an option is specified in more than one way, the system uses the last one specified.

Using the PARM field to specify options

You can specify run-time options with the optional PARM field on the EXEC statement in the NPF FSS writer start procedure. Separate each option by a comma. For example:

```
//IEFPROC  EXEC  PGM=EZAPPFS,REGION=3M,PARM='ESTAE-SDUMP=NO'
//STEPLIB DD DSN=TCPIP.V3R2.LOAD,DISP=SHR
```

Using a PARMLIB DD statement to specify options

You can specify run-time options with an optional PARMLIB DD statement in the NPF FSS writer start procedure.

This statement points to a sequential data set or partitioned data set member. It must contain fixed 80-byte records (RECFM=F or FB, LRECL=80). Each record in this data set can be a string of options from 1 to 80 bytes long. Each option should be separated by a comma with no blanks between. Options must start in column 1.

For example,

```
       //IEFPROC   EXEC   PGM=EZAPPFS,REGION=3M
       //PARMLIB   DD     DSN=USER1.SOURCE(INPARM),DISP=SHR
```

```
//STEPLIB   DD     DSN=TCPIP.V3R2.LOAD,DISP=SHR
  ...
```

where USER1.SOURCE(INPARM) might contain:

```
ESTAE-SDUMP=NO
SPIN=GROUP
```

Using the MVS modify command to change options

After the NPF FSS writer has been started, you can dynamically make changes by passing an options string with the MVS MODIFY command.

For example, if the FSS was defined as TCPFSS, at the operator console you can enter:

```
F TCPFSS,ESTAE-SDUMP=NO
```

When using the MODIFY command, be aware of when the new option takes effect. Some options take effect immediately, others take effect on subsequent print jobs. Some options are captured when an FSA or device is started. In these cases, that FSA would have to be stopped and restarted to cause the new options to take effect.

Defining, starting, and stopping the NPF FSS writer

Put your short description here; used for first paragraph and abstract.

Table 13 on page 115 describes how to define, start, and stop the NPF FSS writer on various levels of JES. Before issuing any of these commands, the procedure to start the NPF FSS writer must be catalogued in a system or recognized PROCLIB. See "Creating the NPF FSS writer start procedure" on page 111.

Because the JES capture point is defined as a functional subsystem, only the JES commands ($SPRTx or *S,PRTx) can be used to start the NPF writer. The MVS START command cannot be used to initiate the writer.

Table 13. Defining, starting, and stopping the NPF FSS writer

<table>
<tr><th>Level</th><th>How to Define</th><th>How to Start or Stop</th></tr>
<tr>
<td>JES2 Version 6</td>
<td>

The functional subsystem and FSS writer definitions can be specified either through JES initialization parameters or by entering commands interactively.

The following example shows the initialization parameters to define an NPF FSS called TCPFSS which points to a start procedure called FSWTR. It also defines two FSS writers called PRT5 and PRT6 that process class D output.

```
(optional:)
DESTID(DEST1) DEST=U1
(optional:)
DESTID(DEST2) DEST=U2

FSS(TCPFSS)  PROC=FSWTR,AUTOSTOP=YES

PRT5 CLASS=D,SEP=YES,
     SEPDS=NO,
     DRAIN,MODE=FSS,
     FSS=TCPFSS,
     TRKCELL=YES,
     PRMODE=(LINE,PAGE),
     NPRO=0,
     ROUTECDE=DEST1,
     SETUP=NOHALT,
     WS=(Q,R/)
```

```
PRT6 CLASS=D,SEP=YES,
     SEPDS=NO,
     DRAIN,MODE=FSS,
     FSS=TCPFSS,
     TRKCELL=YES,
     PRMODE=(LINE,PAGE),
     NPRO=0,
     ROUTECDE=DEST2,
     SETUP=NOHALT,
     WS=(Q,R/)
```

The following example shows the console commands to define an NPF FSS called TCPFSS which points to a start procedure called FSWTR and to define two FSS writers to process class D output.

```
$ADD FSS(TCPFSS),
     PROC=FSWTR,AUTOSTOP=YES
$ADD PRT5,MODE=FSS,
     FSS=TCPFSS,CLASS=D
$ADD PRT6,MODE=FSS,
     FSS=TCPFSS,CLASS=D
```

</td>
<td>

To start the FSS writer that is processing class D output, from the operator console, enter:

```
$SPRT5
```

To stop the same FSS writer, from the operator console, enter:

```
$PPRT5
```

To start both FSS writers, two commands are needed. From the operator console, enter:

```
$SPRT5
```

```
Then enter:
$SPRT6
```

To stop both FSS writers, enter:

```
$PPRT5
```

```
Then enter:
$PPRT6
```

</td>
</tr>
</table>

<table>
<caption><em>Table 13. Defining, starting, and stopping the NPF FSS writer (continued)</em></caption>
<tr><th>Level</th><th>How to Define</th><th>How to Start or Stop</th></tr>
<tr>
<td>JES3 Version 6</td>
<td>The functional subsystem and FSS writers are defined by JES initialization parameters.
The following example shows the initialization parameters to define an NPF FSS called TCPFSS which points to a start procedure called FSWTR. It also defines an FSS writer called PRT501 that processes class C output.
<pre>
DEVICE,DTYPE=PRT3820,
       JNAME=PRT501,
       DYNAMIC=NO,
       MODE=FSS,
       FSSNAME=TCPFSS,
       HEADER=YES,
       BURST=NO,
       NPRO=NO,
       SETUPMSG=NO,
       FORMS=(YES,STANDARD),
       JUNIT=(,SY1,S1,ON,,SY2,S2,OFF),
       PM=LINE,WS=(CL,F),WC=C
FSSDEF,TYPE=WTR,
       FSSNAME=TCPFSS,
       SYSTEM=(SY1),
       PNAME=FSWTR,
       TERM=YES,AUTOSTOP=YES
</pre></td>
<td>To call the FSS writer that is processing class C output, from the operator console, enter:
<pre>
*X,WTR,OUT=PRT501
</pre>
To start the FSS writer that is processing class C output, from the operator console, enter:
<pre>
*S,PRT501
</pre>
To enable the FSS writer to process class D output instead of class C, from the operator console, enter:
<pre>
*X,WTR,OUT=PRT501,WC=D
</pre>
To stop the same FSS writer, from the operator console, enter:
<pre>
*C,PRT501
</pre></td>
</tr>
</table>

For more information about starting and stopping FSS writers, see z/OS JES2 Commands or z/OS JES3 Commands.

Helpful reminders

- JES output files can be routed by the following methods:
  - Specify the CLASS, DEST, and FORMS parameters in the JCL of the print jobs before you submit them.
  - With JES2, use the Spool Display and Search Facility (SDSF) to dynamically alter CLASS, FORMS, or DEST specifications of jobs already in the queue.
- IBM recommends using the NPRO=0 or NPRO=NO in your FSS definitions. If you use any other value, the NPF FSS writer processes the last file on the JES queue and leaves it on the queue until either a new file enters the queue or an internal timer runs out.
- If your JES3 installation is set up to route all output to the JES hold queue, you can use the following commands to move a job from the hold queue to the writer queue:

```
     *I,J=jobname                     to find out the job number
     *F,U,J=jobnumber,Q=HOLD,NQ=WTR   to move the job from the JES
                                      hold queue to the NPF FSS writer
                                      queue
```

- The following example shows you how to set up an NPF FSS writer for called PRT6 that uses CLASS as the only selection criterion. See *JES2 Commands* or *JES3 Commands* for variations that might be more appropriate for your specific environment.

```
    $DU PRT6          to display the printer setup
    $T PRT6,WS=(-R/)   PRT6 uses CLASS as the only selection criterion
```

- JES2 users should use the SEP parameter in their FSS writer definitions. SEP is the default, and is specified as follows:

- For JES2 Version 4, specify SEP=YES
- For JES2 Version 2 and 3, specify SEP NOSEP, available in JES2 Version 2 and 3, will group more than 2 data sets into 1.
- We recommend that the AUTOSTOP=YES parameter be specified on the FSS definition.

Operating the NPF VTAM capture point

The following sections show you the JCL needed to start the NPF VTAM capture point application, the commands you can use to modify it while it is running, and the options you have for stopping it.

Starting the NPF VTAM capture point

The following shows a sample procedure to start an NPF VTAM capture point application. You can copy this sample from *hlq*.SEZAINST(EZAPPAAA) and modify it to suit your environment.

Note: The log file cannot be shared. If you want to run more than one instance of the VTAM capture point application, you will need separate startup procedures with a unique name for each data set in each procedure.

```
//EZAPPAAA JOB
//*--------------------------------------------------------------------*
//*
//* COPYRIGHT = NONE
//*
//* SMP/E Distribution Name: EZAEC0YS
//*
//* This JCL starts the Network Print Facility's VTAM Capture Point
//* application.
//*
//*   The executable code is assumed to be in a link library which is
//*   in the system link list.  (The default is TCPIP.SEZALINK).
//*
//*   In this sample, below-the-line storage usage is assumed to have
//*   been estimated as follows:
//*      0.5M  - for a minimum-sized application
//*      3.5M  - for storage to handle in-process print files.
//*
//*   The following values in the JCL reflect these assumptions:
//*
//*      The REGION parameter specifies the maximum amount of
//*      below-the-line storage which MVS will allow for this
//*      application.  The 4M value in this sample is the sum of the
//*      two assumed values described above.
//*
//*      The MAXFLSTG subparameter limits the amount of below-the-line
//*      storage which the Network Print Facility will request for
//*      print files which are still in process of being received.
//*      Each open print file uses up 16K of this limit.  The 3584K
//*      value (3.5M) specified in this sample will allow a maximum
//*      of 224 print files to be open at the same time.
//*
//*   TIME is coded with 1440 so we don't time out.
//*
//*   The subparameter LUCLASS is coded (1,2) to pick up any printers
//*   in our LU classes 1 and/or 2.
//*
//*   In this sample, the names of the routing, options and queue
//*   files are:
//*
//*       TCPIP.ROUTING
//*       TCPIP.OPTIONS
//*       TCPIP.QUEUE
//*
//*   All three are assumed to have been created and initialized
//*   prior to the use of this JCL.
//*
//*   The log data set for this sample is TCPIP.VTAM1.LOG.
//*
//*   NOTE: All temporary data sets created by Network Print Facility
//*         will have the high-level qualifier specified via the
//*         defined search sequence.  (See "Configuring NPF Using
//*         TCPIP.DATA and NPF.DATA Statements" in IBM TCPIP for MVS:
//*         Network Print Facility).
//*
//*         In this sample, user specified defaults will be found
//*         in the TCPIP.DATA data set defined by the SYSTCPD DD
//*         statement and in the NPF.DATA data set defined by the
//*         SYSNPFD DD statement.
//*
//*-------------------------------------------------------------------*
//EZAPPAAA EXEC PGM=EZAPPAAA,REGION=4M,TIME=1440,
//         PARM='LUCLASS=(1,2),MAXFLSTG=3584K'
//EZAPPRF  DD DSN=TCPIP.ROUTING,DISP=SHR
//EZAPPOF  DD DSN=TCPIP.OPTIONS,DISP=SHR
//EZAPPQF  DD DSN=TCPIP.QUEUE,DISP=SHR
//EZAPPLOG DD DSN=TCPIP.VTAM1.LOG,DISP=OLD
//SYSTCPD  DD DSN=TCPIP.TCPIP.DATA,DISP=SHR
//SYSNPFD  DD DSN=TCPIP.NPF.DATA,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSMDUMP DD SYSOUT=*
```

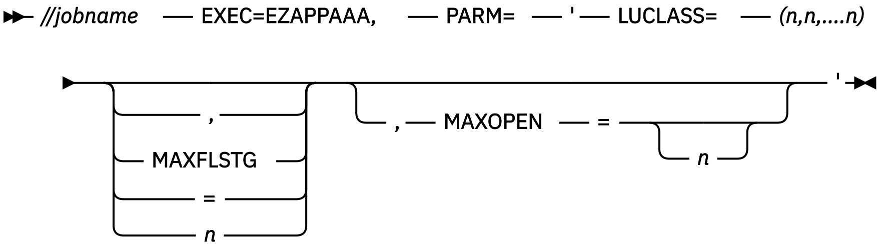
Specifying VTAM capture point parameters

The required and optional parameters for the VTAM capture point application are passed by the PARM parameter on the EXEC statement of the start procedure. Add your parameters following PARM=, making certain that:

- All parameters are in uppercase
- Parameters are separated by commas
- Each set of parameters is enclosed in quotes

For example: `//EZAPPAAA EXEC PGM=EZAPPAAA,PARM='LUCLASS=(1,2),MAXFLSTG=2150K'`

Syntax



Parameters

LUCLASS=*(n,n,...n)*

Specifies the LU names from the routing file that this instance of the Network Print Facility's VTAM capture point will process.

This field is required. Valid values are 1 to 64. Separate each value with a comma. No comma should follow the last value.

MAXFLSTG=*n*

Specifies the maximum amount of below-the-line private storage you want to allow for the processing of open temporary print data sets.

This parameter is optional and has no default value. The value specified is limited to 8 characters and can specify a number of bytes, kilobytes or megabytes. For example, you can specify a value of 2 megabytes as 2097152, 2048K or 2M.

This parameter is intended to prevent abends caused by excessive requests for below-the-line storage. The value should equal the REGION parameter value minus the total below-the-line storage required for code and control blocks. See “Creating a startup procedure” on page 22.

When the below-the-line storage needed for a new temporary print data set would cause the MAXFLSTG limit to be exceeded, the new print data request is rejected with an X'08120000' (insufficient resources) sense code. This prevents a GETMAIN request which could cause an abend due to a REGION size violation.

MAXOPEN=*n*

Specifies the maximum number of temporary print data sets that can be open at any given time.

At 16 KB of below-the-line storage per open print data set, this is simply a less direct, alternative method of specifying the same storage limit described under MAXFLSTG.

This parameter is optional and has no default value. If the parameter is omitted, a maximum of 1 temporary print data set per logical printer can be open at any given time.

The parameter value is limited to 8 digits and must be non-zero.

When the opening of a new temporary print data set would cause the MAXOPEN limit to be exceeded, the new print data request is rejected with an X'08120000' ('insufficient resources') sense code.

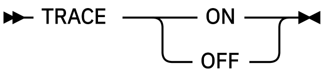
Modifying the NPF VTAM capture point

While the VTAM capture point application is running, you can dynamically add or delete logical printers and replace the end-of-file rules and default page format table. To do this, at the operator's console, enter LUNAME or RELOAD in response to message EZY0837I. You can also dynamically turn debug tracing on or off in response to EZY0837I WTOR.

TRACE: Turn debug tracing on or off

Use the TRACE operator response to dynamically turn debug tracing on or off in response to EZY0837I WTOR.

Syntax



Parameters

ON

Turns on debug tracing for the VTAM capture point. The bulk of trace entries are WTOs with ROUTECDE=(11). They go to the JES message log for the VTAM capture point job and to the SYSLOG. Normally, the main operator console is not set up to receive any "programmer information" type Route Code 11 WTOs. TRACE ON will also SNAP out data to the EZAPPSNP DD if it is added to the startup procedure for the VTAM capture point EZAPPAAA. This DD is similar to EZAPPLOG and can be defined to go to SYSOUT=*.

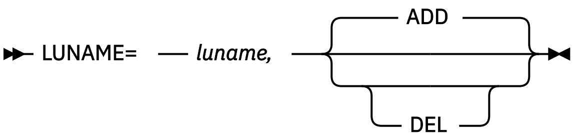
OFF

Turns off debug tracing for the VTAM capture point. The VTAM capture point debugging trace is normally off. After the trace has been dynamically started using the TRACE ON reply to the EZY0837I message, you can turn it off with the TRACE OFF command.

LUNAME: Adding or deleting logical printers

Use the LUNAME operator response to dynamically add or delete a printer from the list of logical printers for this NPF VTAM capture point.

Syntax



Parameters

luname
The 8-character logical printer to add or delete. This name is the same as the major name key field in the ROUTING file.

ADD
Adds a new printer. This *luname* must already have a VTAM entry in the routing file (one that has an LUCLASS specified).

DEL
Deletes an existing printer. The *luname* must already be inactive in the NPF's VTAM capture point. This can be done by deactivating the LU in VTAM. If the logical printer was never active in NPF's VTAM capture point (it was unable to be opened), then it can be deleted at any time.

Usage notes

The new logical printer is considered added after the ROUTING file entry is located, related control blocks are created, and an initial attempt to OPEN its ACB is complete. For the printer to become active, it must also already be defined and active in VTAM.

Examples

To add a logical printer to the NPF's VTAM capture point, enter this response to message EZY0837I:

```
R nn,LUNAME=ELU10100,ADD
```

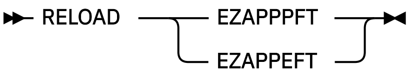
To delete a logical printer from the NPF's VTAM capture point, enter this response to message EZY0837I:

```
R nn,LUNAME=ELU10100,DEL
```

RELOAD: Dynamically replacing a table

Use the RELOAD operator response to dynamically replace the end-of-file rules or default page format tables. This operator response does not affect any current session, only sessions established after executing this command.

Syntax



Parameters

EZAPPPFT
Replaces the default page format table

EZAPPEFT
Replaces the end-of-file rules table

Examples

To replace the default page format table, enter this response to message EZY0837I:

```
R nn,RELOAD EZAPPPFT
```

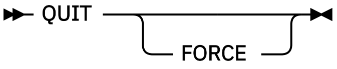
Stopping the NPF VTAM capture point

You can terminate the VTAM capture point application by issuing a QUIT, QUIT FORCE, or KILL response to message EZY0837I.

QUIT: Normal termination

Use one of the variations of the QUIT command as the normal way to terminate NPF's VTAM capture point application.

Syntax



QUIT
Prevents any new sessions from starting but allows each current session to continue normal operation. Only after all current sessions have ended naturally will the NPF VTAM capture point terminate.

Because this request does nothing to force an end to current sessions, it is possible to have a long delay between entering the reply and the actual termination. Each session ends naturally only when the session partner LU decides to end it.

QUIT FORCE
Acts the same as QUIT, except that current sessions are forced to end immediately instead of being allowed to continue. QUIT FORCE normally terminates more quickly than QUIT; however, any files only partially received when QUIT FORCE is issued will result in output with incomplete data.

Examples

To stop the NPF's VTAM capture point, enter this response to message EZY0837I:

```
   R nn,QUIT
```

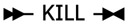
To stop the NPF's VTAM capture point and force all current sessions to end, enter this response to message EZY0837I:

```
   R nn,QUIT FORCE
```

KILL: Immediate Termination

Use the KILL operator response only for emergency termination of NPF's VTAM capture point. This request causes immediate termination with no attempt to end sessions or clean up open print files. Loss of print files can occur.

Syntax



Attention: This is not a recommended operator control. It is included only to allow the operator a way out if a hang condition prevents completion of a QUIT FORCE request.

Examples

To stop the NPF's VTAM capture point and all sessions immediately, enter this response to message EZY0837I:

```
R nn,KILL
```

Effects of VTAM HALT commands on the NPF VTAM capture point

When a console operator enters a HALT command to end VTAM, the effect to the NPF VTAM capture point application depends on the type of HALT.

- A standard HALT causes the orderly closing of the ACBs for all logical printers, but the Network Print Facility continues.

  No new sessions are allowed to start, but existing sessions are allowed to continue normally. After a logical printer's sessions have ended naturally, the logical printer's ACB is closed. The Network Print Facility application remains operational, periodically attempting to reopen the ACBs for its logical printers.
- A HALT QUICK causes processing essentially identical to that for the standard HALT, except that each existing session is ended immediately via a TERMSESS macro.
- A HALT CANCEL causes the immediate closing of each logical printer's ACB, automatically ending any existing sessions. Data already received is processed. The then ends by returning to MVS.

Operating the queue manager

The following sections show you the JCL needed to start the NPF queue manager, the commands you can use to modify it while it is running, and the options you have for stopping it.

Starting the NPF queue manager

The queue manager program, EZAPPQSA, runs in a separate address space from other Network Print Facility activities.

It is possible to run multiple instances of the NPF queue manager at the same time, subject to the following restrictions:

- Each instance of the NPF queue manager should do processing for a different queue file and should have its own unique log file.
- All queue records for any given printer should be in a single queue file and, therefore, processed by a single instance of the NPF queue manager.

Specifying NPF queue manager parameters

The required parameters for the NPF queue manager are passed by the PARM parameter on the EXEC statement of the start procedure.

Syntax

```
►► // ── jobname ── EXEC=EZAPPQSA, ── PARM= ── ' ── time ── ' ─►◄
```

Parameters

time

This parameter is used to specify the scan interval used by the queue manager. This field is required.

The interval is specified in hours, minutes, and seconds in the format of *hhmmss*. Significant digits are required to be specified. For example, 2 minutes can be coded as 000200 or 200. The maximum value you can enter is 999999.

Note: A long scan interval uses fewer resources but delays the initial attempts to send new print jobs.

Modifying the NPF queue manager

While the NPF queue manager is running, you can dynamically change the scan interval and turn the trace facility on or off. To do this, at the operator’s console, enter INTERVAL, TRACE, or NOTRACE in response to message EZY0960I.

INTERVAL: Change scan time

Use the INTERVAL operator response to dynamically change the scan time interval.

Syntax

```
▶▶ INTERVAL= ── time ─▶◀
```

time

This parameter is used to specify the new scan interval. The interval is specified in hours, minutes, and seconds in the format of *hhmmss*. Significant digits are required to be specified. For example, 2 minutes can be coded as 000200 or 200. The maximum value you can enter is 999999.

Note: A long scan interval uses fewer resources but delays the initial attempts to send new print jobs.

Examples

To change the queue manager's scan time interval to 1 hour and 30 minutes, enter this response to message EZY0960I:

```
R nn,INTERVAL=13000
```

TRACE: Turn on console trace

Use the TRACE operator response to dynamically turn on the console trace. This allows the operator to follow the flow of the NPF queue manager on the console.

Syntax

```
▶▶ TRACE ─▶◀
```

Examples

To start a trace of the NPF queue manager, enter this response to message EZY0960I:

```
R nn,TRACE
```

NOTRACE: Turn off console trace

Use the NOTRACE operator response to dynamically turn off the console trace. This allows the operator to stop the NPF queue manager from reporting on its flow.

Syntax

```
▶▶ NOTRACE ─▶◀
```

Examples

To stop a trace of the NPF queue manager, enter this response to message EZY0960I:

```
R nn,NOTRACE
```

Stopping the NPF queue manager

You can terminate the NPF queue manager by issuing a STOP response to message EZY0960I. This response terminates the queue manager immediately following the completion all its outstanding processing.

For example, to stop the NPF queue manager, enter this response to message EZY0960I:

```
R nn,STOP
```

Stopping the Network Print Facility

In order to stop the Network Print Facility you must stop each of its programs:

- The JES capture point (See Table 13 on page 115)
- The VTAM capture point application (See “Stopping the NPF VTAM capture point” on page 121)
- The NPF queue manager (See “Stopping the NPF queue manager” on page 124)

Because these all are independent programs and run in separate address spaces, they can be stopped individually at any time.

Chapter 9. Diagnosing Network Print Facility Problems

Diagnosis information includes:

- NPF FSS Writer Return Codes
- Lists of Common Problems
- Network Print Facility Tracing
- Generation of the Queue File Name
- SNA Sense Codes

NPF FSS writer return codes

These return codes can be received via Network Print Facility messages EZA0932E and EZA0933E.

```
      0001         Incorrect function code
      0002         Error on program GETMAIN
      0003         Load failure
      0004         Load failure for input record exit
      0005         ATTACH failed
      0006         LINK of LPR failed
      0007         SOKETS initialization failed
      1001         Error on GETMAIN in ALLOC function
      1002         Routing/Options file error
      1003         Open failed
      1004         Routing record not found
      1005         Link error on specific routing exit
      1006         Options record not found
      2001         Close failed
      3001         Reposition input to 1st record for PUT
      3002         PUT failed
      3003         Error from input record exit
      3004         Length error on PUT
      3005         Insufficient space for data file
      3006         PUT attempted on file that is not open
      4001         File state incorrect for SEND
      4002         SEND failed
      4003         Deallocation failed
      4004         PUT Queue error
```

Figure 41. NPF FSS writer return codes

List of common problems and suggested solutions

Table 14. List of common problems for general Network Print Facility processing

Problem	Explanation/Action
You are in a wait state.	Ensure TCPIPJOBNAME is defined.
You receive message CSV003I, `Module EZAPPGPR not found`.	Ignore the message. Anyone not specifying a general routing exit receives this message.

Table 15. List of common problems for processing JES output

Problem	Explanation/Action
You receive a message regarding incorrect routing. It might be that a file is being sent back to the JES hold queue due to invalid routing information (nonexistent printer information, for example).	The file stays there until the user releases or reroutes it. Ensure the destination you specified in your JCL is the same as the destination value listed in the SDSF output display panel. If you are running JES2 Version 3, the destination name must be equal to the host name. If it is not the same, use what is listed in the SDSF output display panel as the destination for your routing file. What is displayed on the SDSF output display panel for class, destination, and forms is what JES is passing to the NPF FSS writer. Refer *JES2 Commands* or *JES3 Commands* for details about how to release held jobs from the JES queue.
The NPF FSS writer issues a user abend due to an unrecoverable JES error condition (not being able to connect to the Network Print Facility, for example).	The file is held on the JES queue until the user releases or reroutes it. All terminating errors caused by the NPF FSS writer interface to or from Network Print Facility routines generate a USER777 abend code. The writer error routine takes a snapshot of the storage before the NPF FSS writer terminates. When the error condition is fixed, the NPF FSS writer can be restarted. In case of invalid routing information (nonexistent printer information, for example), the Network Print Facility issues message EZY0912I indicating that the file is being sent back to the JES queue to be held. See *JES2 Commands* or *JES3 Commands* for details about how to release held jobs from the JES queue.

Network Print Facility tracing

You can run traces of ISPF function using the options available on the NPF ISPF Primary Option Menu or you can run a trace of the queue management function through the use of an operator command.

Using the NPF ISPF trace facility

The Network Print Facility ISPF trace is useful for internal debugging only. Using this trace without the support of IBM service personnel might not be productive. You can use the NPF ISPF trace to trace an ISPF function (add or delete, for example). Information captured can then be displayed in a data set for evaluation and problem resolution.

Procedure

Follow these steps to implement the NPF ISPF trace:

1. Optional: Create the sequential data set you want to use for tracing with these specifications: sequential organization (PS), a variable block format (VB), a logical record length (LRECL) of 132, and any block size (BLKSIZE).
2. On the NPF Diagnosis Functions panel (EZAPPDGP), specify this data set or enter a data set name that begins with your TSO user ID and the system will allocate one for you. (See Figure 15 on page 69.)
3. On the same panel, enter a **2** to turn tracing on.

4. After the operation is complete, return to the NPF Diagnosis Functions panel enter a **1** to turn tracing off.
5. IBM service personnel will inspect the trace data set. Information will be available on data input to the operation and data output to the operation. Evaluate the data to ensure the names match, and appropriate fields are filled-in.

Results

Note: The trace facility is used only to trace the path of input to and output from NPF ISPF panels. It is not intended to be used to trace an abend. For example, if the TRACE is on and an abend occurs, you might not get complete data.

Using the queue manager trace

The operator can specify a trace of the queue management function by issuing the following commands:

```
R xx,TRACE               to turn on console trace
R xx,NOTRACE             to turn off console trace
```

The queue manager writes a message to the console for each file action, including retry attempts or file deletions. See the explanation of message EZY0960I in the z/OS Communications Server: IP Messages Volume 3 (EZY) manual. This message describes the routing, name of file, type of operation, and time. The time displayed is the time the scan started, not the time the actual operation occurs.

The trace output goes to the MVS console and the normal job log specified in the NPF start procedures (either NPF FSS writer (for JES) or VTAM).

Using the VTAM capture point trace

The operator can specify a debugging trace of the VTAM capture point by issuing the following commands in reply to the EZY0837I message (WTOR):

```
R xx,TRACE               to turn on console trace
R xx,NOTRACE             to turn off console trace
```

The trace entries are "programmer information" type WTOs with ROUTCDE=(11). They go to the JES message log for the VTAM capture point job and to the SYSLOG, but they are not displayed on the main operator console unless it has been configured to accept ROUTCDE(11) messages.

Additionally, if an EZAPPSNP DD is provided in the startup procedure for EZAPPAAA, then additional debugging information will be written to that DD. The EZAPPSNP DD is similar to the EZAPPLOG DD and can be SYSOUT=* or a sequential data set(PS) with RECFM=FBA, LRECL=133, and BLKSIZE=1330. If a sequential data set is used, be sure to provide enough space.

Network Print Facility logging

The Network Print Facility has a logging facility that writes messages in a log file to track the system activity regarding temporary print data sets. The messages (EZY0970I, EZY0971I, EZY0972E, EZY0973I) record when a print data set is placed on the queue, successfully or unsuccessfully transmitted, and deleted.

For logging, create a QSAM data set with VBA or FBA format, a logical record length (LRECL) of 133, and any blocksize.

The log file is specified with the //EZAPPLOG DD statement in the JCL to start the capture point applications and queue manager. The log file cannot be shared. Therefore, you must have a unique log file for each startup procedure for these programs. If the //EZAPPLOG statement is missing, the system will not log this activity.

Generation of the print data set names

Network Print Facility's queue file manager creates a name for each data set it tracks. The Network Print Facility derives the names for the temporary print data sets as follows:

- The name is divided into 4 sections:
  - High-level qualifier
  - Major Key (JES DEST or SLU name)
  - Minor Key (JES job name or VTAM PLU)
  - Date and time
- NPF obtains the high-level qualifier from an NPFPRINTPREFIX configuration statement, if one is specified. If not, NPF uses the TCP/IP high-level qualifier.
- NPF strips the major key of any trailing blanks.
- NPF strips the minor key of any trailing blanks.
- NPF derives the date and time as follows:
  - Calculates the modulo 25 value for the year, expressed as a letter A=0, B=1,.....,Y=24.
  - Expresses the julian day as a 3-digit number.
  - Expresses the hour in base25 using letters as in the year.
  - Expresses the minute as a 2-digit number.
  - Inserts a period because this makes up 7 characters.
  - Expresses the seconds and tenths of seconds as a 2-digit base 25 number using letters as above. Thus the date and time is `YdddHmm.SS` where the upper case letters represent the modulo/base 25 letters and the lower case letters represent numbers,
- NPF starts the name with the complete high-level qualifier.
- NPF ends the name with the complete date.
- NPF merges the lengths of the major and minor keys into one string if the sum is <= 8. Otherwise, NPF uses them as two strings separated by a period.
- If necessary, NPF truncates the major/minor keys from the right to fit within the 44-character limit required by MVS. If the length of the high-level qualifier is 14 or less, no truncation occurs.

SNA sense codes used by the NPF VTAM capture point

An error on an input request is indicated by the return of either a negative response or (for BIND only) an UNBIND request. In either case, the error indication includes a sense code.

Sense codes used by the NPF VTAM application program fall into two categories:

- General-use sense codes already described in *SNA - Sessions Between Logical Units*. These are not discussed here.
- Implementation-specific sense codes, which are not described in standard SNA documentation. These sense codes and their meanings are described in the tables that follow.

Table 16. SNA sense code for BIND. These sense codes are sent on a BIND response or UNBIND request when the Network Print Facility rejects a BIND.

Sense Code	Meaning
X'081C0190'	Request Not Executable Network Print Facility could not find the specified entry in the end-of-file rules table.
X'081C0191'	Request Not Executable Network Print Facility could not find the specified entry in the page format table.

Table 17. SNA sense codes for FM data. These sense codes are sent on the response to an FM data (normal print data) request.

Sense Code	Meaning
X'081C0002'	Request Not Executable Internal logic error detected in SCS logical printer.
X'081C0100'	Request Not Executable A LOAD for the input record exit failed. Network Print Facility software error.
X'081C0101'	Request Not Executable A LINK to LPR failed.
X'081C0102'	Request Not Executable The attempt to allocate a QSAM file failed.
X'081C0103'	Request Not Executable An error has been found in the routing or options file.
X'081C0104'	Request Not Executable The attempt to open a QSAM file failed.
X'081C0105'	Request Not Executable No routing record was found for the logical printer.
X'081C0106'	Request Not Executable The attempt to close a QSAM file failed.
X'081C0107'	Request Not Executable The input record exit specified an action valid only with JES (reposition input to the first record for the next put).
X'081C0108'	Request Not Executable The attempt to put the request's print data into a QSAM file failed.
X'081C0109'	Request Not Executable An error occurred in the input record exit.
X'081C0200'	Request Not Executable A length error occurred while trying to put the request's print data into a QSAM file.
X'081C0201'	Request Not Executable The deallocation of a QSAM file failed.
X'081C0202'	Request Not Executable The attempt to put the request's print data into a QSAM file failed.
X'081C0203'	Request Not Executable An unexpected return code was returned.
X'081C0220'	Request Not Executable Error detected in Page Format Table entry for this session. Reload a correct EZAPPPFT or modify the routing file to use a correct Page Format Table entry.

Appendix A. Checklists for using the Network Print Facility

Use the 3 checklists provided here to plan for, customize, and use the Network Print Facility.

Checklist of the NPF tasks for all users (JES or VTAM)

- Decide how many logical printers you will use.
- Decide how you can effectively implement the routes you require. Should jobs be sent to one printer or to multiple homogeneous printers?
- Decide how you will use the LPR print options. Which will be the system default options and which will be specified in your options file? Plan to use the OPTIONS function of the NPF EZAPPFL macro or the NPF ISPF interface to create the options records.

  See the z/OS Communications Server: IP User's Guide and Commands for a complete description of LPR options.
- Establish your own job header and trailer through LPR definitions; NPF does not provide job header and trailer options itself. If desired, use the IBM-supplied separator or banner defaults, provided with LPR. See the LPR documentation in the z/OS Communications Server: IP User's Guide and Commands.
- Decide on unique major and minor names for your print routings. Plan to use the ROUTING function of the NPF EZAPPFL macro or the NPF ISPF interface to create the routing records.
- Define the queue management function.
  - Decide how to handle output that ***does*** print successfully. Do you want to delete the job immediately? Do you want to retain the job for a period of time for future use? Plan to use the RETAINS option on the ROUTING function of the NPF EZAPPFL macro or the ISPF panel interface to specify your choices.
  - Decide how to handle output that ***does not*** print successfully. Do you want to delete the job immediately? Do you want to retain the job for a period of time for future use? Do you want to resend the data? If so, how many times do you want to attempt to resend? At what interval? Plan to use the RETAINU and RETRY options on the ROUTING function of the NPF EZAPPFL macro or the ISPF interface to specify your choices.
- Determine if user exits are necessary. If so, design, code, and install them. Plan to specify them appropriately in the macros, data areas, parameter lists, and in the routing and options records.

  See Chapter 7, “Writing exit routines to tailor the Network Print Facility,” on page 93.
- Install the C run-time libraries to implement the ISPF interface. This will require a re-IPL to make the change permanent.
- Configure your system for NPF by using the appropriate configuration statements.

  See “Configuring NPF using TCPIP.DATA and NPF.DATA statements” on page 15.
- Assign appropriate user authorization. This might relate to the alteration of your ISPF logon procedure to enable NPF.

  See the appropriate RACF manual or other security program publication.
- Enable the NPF ISPF interface.
  - Provide library access
  - Update the ISPF primary menu to include an option to access NPF.
  - Establish optional default values for the routing, options, queue, and trace files to display on the interface panels.

  See “Enabling the Network Print Facility ISPF interface” on page 59.

- Modify the program properties table with the JES PPT entry or the VTAM PPT entry or both (see the checklist specific to JES or VTAM for the exact update). Modifying the PPT requires a re-IPL.

  See the z/OS MVS Initialization and Tuning Guide for information about updating the PPT.
- Define the JES capture point (NPF FSS writers) or VTAM capture point (applications), or both. See the Checklist for JES users, or the Checklist for VTAM users, on the following pages.
- Allocate and initialize the VSAM files for routing, option, and queue files.

  See Chapter 5, “Creating the Network Print Facility files,” on page 43.
- Load the routing and options file records using the various NPF EZAPPFL macro functions.

  See Chapter 5, “Creating the Network Print Facility files,” on page 43.
- For each different VTAM capture point, JES capture point, or queue manager application, allocate a QSAM data set for the LOG file.

  See “Network Print Facility logging” on page 127.
- Start either the NPF FSS writer to process output from JES or the VTAM capture point to process output from VTAM, or both. Start the NPF queue manager to handle the retention and retransmission of print data sets. These require you to submit the JCL start procedures for each.

  See Chapter 8, “Operating the Network Print Facility,” on page 111
- Maintain your files by changing print options, routing destinations, and data fields that affect the disposition of print data sets. Use either the ISPF panel interface or the EZAPPFL macro.

  See Chapter 5, “Creating the Network Print Facility files,” on page 43 and Chapter 6, “Maintaining the Network Print Facility files with ISPF,” on page 59.

  Note: Using the EZAPPFL macro updates the VSAM files in batch mode while ISPF performs the same function interactively. When large numbers of updates are required, using the EZAPPFL macro might be more efficient.
- If a problem is encountered using the NPF panel interface, use the NPF ISPF trace to find inconsistencies in panel input and output.

  See “Using the NPF ISPF trace facility” on page 126.

Checklist of the NPF tasks for VTAM users

- Establish consistent definition of resources.
  - Set up an APPL definition for each logical printer name. Define each of the logical printer (SLU) names to VTAM by an APPL definition statement in an application major node definition.

    See the z/OS Communications Server: SNA Resource Definition Reference.
  - In the partner LU, set up definitions for each logical printer (SLU) name. For CICS, a specific requirement is that you specify NO for the QUERY keyword on the TYPETERM macro.

    See IMS or CICS publications relating to resource definition.
  - Build the routing file to make the connection between the logical printer names and the LUCLASS numbers. See “Routing file input fields” on page 39.
- Run nonswappable. Add a Network Print Facility entry for the VTAM application, EZAPPAAA, in the MVS program properties table (PPT). The EZAPPAAA entry must be specified as: `PPT PGMNAME(EZAPPAAA)`

  Note: PPT entries in the SCHED*xx* member of SYS1.PARMLIB are no longer required. They are supplied in the default PPT table via load module IEFSDPPT.
- Estimate the appropriate region size for the job to start the Network Print Facility. The size depends on the maximum number of output files in progress at any given time, the number of logical printers, and the number of sessions.
- Set up the JCL to run the Network Print Facility for the VTAM capture point. There are four major types of information that this JCL must specify:

1. On the EXEC statement, code PGM=EZAPPAAA to identify the program to be started.
2. Also on the EXEC statement, code the PARM parameter with the LUCLASS subparameter to specify one or more LU class numbers with values from 1 to 64 and the REGION and MAXFLSTG parameters to accommodate your below-the-line storage estimates.
3. Code DD statements for the routing, options, queue, and log files. The ddnames must be coded exactly as shown in the VTAM start sample. The system administrator can choose the data set names.
4. On the JOB statement, code `TIME=1440` or `TIME=NOLIMIT` to prevent MVS from terminating the application because of its running time.

   See "Starting the NPF VTAM capture point" on page 117.

- For end-of-file processing:

  1. Evaluate if IBM's default end-of-file rule is sufficient for your installation. See "Defining end-of-file rules" on page 27.
  2. If necessary, use the EZAPPEFM macro to modify the end-of-file rules table (EZAPPEFT), assemble and link-edit it into a STEPLIB available to the NPF VTAM application or the link list. See "Defining end-of-file rules" on page 27.
  3. Specify the optional end-of-file rules table entry name in the SLU's routing file record. Several SLUs can share the same table entry. The Network Print Facility uses DFLTNTRY if you do not specify an entry name. See "EZAPPFL TYPE=ROUTING" on page 45.

- For page format processing:

  1. Evaluate if IBM's default page format is sufficient for your installation. See "EZAPPDPF" on page 34.
  2. If necessary, use the EZAPPDPF macro to create the page format table (EZAPPPFT), assemble and link-edit it into a STEPLIB available to the NPF VTAM application. See "EZAPPDPF" on page 34.
  3. Specify the optional page format table entry name in SLU's the routing file record. Several SLUs can share the same default page format. The Network Print Facility does not require any specification. See "EZAPPFL TYPE=ROUTING" on page 45.
  4. Dynamically replace the page format table, using the RELOAD command. See "RELOAD: Dynamically replacing a table" on page 120.

- Start the VTAM capture point. See "Starting the NPF VTAM capture point" on page 117.
- Tune the VTAM capture point application by adjusting the limits you specify in the start procedure. See the **Tuning note** in "Creating a startup procedure" on page 22.

Checklist of the NPF tasks for JES users

- Decide how many NPF FSS writers or how many FSA subtasks you will use.

  See z/OS MVS Using the Functional Subsystem Interface.

- Determine where to print JES output.

  See the z/OS MVS JCL Reference.

- Decide how you can most effectively use the JCL CLASS parameter. If necessary, decide what selection criteria you will use in addition to CLASS to send output to an NPF FSS writer.

  See z/OS JES2 Commands or z/OS JES3 Commands for examples of the $T command, or see "Determining where to print JES output" on page 21 in this book.

- Evaluate JESNEWS affect on the Network Print Facility.

  See z/OS JES3 Commands or z/OS JES2 Initialization and Tuning Guide, or see "Customizing JESNEWS" on page 21 in this manual.

- Authorize the NPF FSS writer load module. For JES2, place EZAPPFS in the MVS program properties table (PPT) (with key=1) to ensure subpools 0–127 are in key 1. The EZAPPFS entry must be specified as follows:

  ```
  PPT  PGMNAME(EZAPPFS) KEY(1)
  ```

  Note: PPT entries in the SCHED*xx* member of SYS1.PARMLIB are no longer required. They are supplied in the default PPT table via load module IEFSDPPT.
- JES3 users should have APAR OY50274 applied in order for Network Print Facility to make use of field JSPJGRPD, which contains the DEST (destination) value.

  See the z/OS Communications Server: New Function Summary.
- Accommodate NPF's maximum 8-character restriction for destination names.

  See Note.
- If you are running JES2 Version 3, ensure the destination name is the same as the TCP/IP or MVS host name.
- Ensure your own changes made to the job separator page area (IAZJSPA) (such as DEST, CLASS, FORMS, job name, and user ID) and extension areas (or changes made in JES2 exit 23 or JES3 exit 45, for examples) are compatible with the NPF FSS writer. (The NPF FSS writer uses some of these fields.)
- Set up the NPF FSS writers you need using JES FSSDEF initialization statements. This includes tailoring the NPF FSS writer to your installation needs.

  See “Defining the NPF FSS writer” on page 21.
- Set up the JCL to start the NPF FSS writer, and start it.

  See “Creating the NPF FSS writer start procedure” on page 111.

Appendix B. Network Print Facility's Routing Data Area (RDA)

The routing data is a combination of data from the routing and options files and is accessible to all three exits. If the general routing exit builds the routing data area, it should initialize all fields. Fields that can be modified by either the specific routing exit or the input record exit are marked.

Field Name
  Field Description

RTDAMAJK
  An 8-byte character field containing the major name for this routing. For JES, this corresponds to the CLASS parameter in the JCL. For VTAM, this corresponds to the LU Name.

  This field should not be modified.

RTDAMINK
  An 8-byte character field containing the minor name for this routing. For JES, this corresponds to the DEST parameter in the JCL. For VTAM, this must contain nonblanks.

  This field should not be modified.

RTDASREX
  An 8-byte character field containing the name of the specific routing exit, if one was specified. If not, this field contains binary zeros.

  This field should not be modified.

RTDAIREX
  An 8-byte character field containing the name of the input record exit if one was specified. If not, this field contains binary zeros.

  This field should not be modified.

RTDARTRS
  A fullword binary field which gives the period of time the system should retain data for this routing after a successful transmission to LPR. This field has the format *dddhhmm*, where:

  ddd
    The number of days to retain the file after successful transmission to LPR.

  hh
    The number of hours to retain the file after successful transmission to LPR.

  mm
    The number of minutes to retain the file after successful transmission to LPR.

  The total time to retain the file is the sum of the days, hours and minutes.

  This field can be modified by either exit.

RTDARTRU
  A fullword binary field which gives the period of time the system should retain data for this routing after an attempting the specified number of retries without success. This field has the format *dddhhmm*, where:

  ddd
    The number of days to retain the file after an unsuccessful transmission to LPR.

  hh
    The number of hours to retain the file after an unsuccessful transmission to LPR.

  mm
    The number of minutes to retain the file after an unsuccessful transmission to LPR.

The total time to retain the file is the sum of the days, hours and minutes.

This field can be modified by either exit.

RTDARTRR

A fullword binary field which gives the period of time the system should wait before attempting a retry on this routing after an unsuccessful attempt to transmit to LPR. This field has the format *dddhhmm*, where:

ddd

The number of days to wait before a retry of an unsuccessful transmission to LPR.

hh

The number of hours to wait before a retry of an unsuccessful transmission to LPR.

mm

The number of minutes to wait before a retry of an unsuccessful transmission to LPR.

The total time to wait before a retry of an unsuccessful transmission to LPR is the sum of the days, hours and minutes.

This field can be modified by either exit.

RTDADEST

A halfword binary number containing the number of destinations described by this routing record. The default is 1 (normal routing). A number greater than 1 means this record represents a specific broadcast.

This field can be modified but caution is advised. The system has no way to determine if it is consistent with the number of destinations described in RTDANAMS.

RTDARTRL

A halfword binary field which specifies the number of retry attempts to be made on this routing. If the file is not successfully transmitted by LPR within this number of retries, it changes to retain status.

This field can be modified by either exit.

RTDATYPE

A 1-byte character field containing the value 'N'. This record represents one or more destinations which have the same characteristics. It might be followed by additional records describing other destinations for the same routing. If the field NO_OF_DEST is 1, the record represents a normal routing. If the field NO_OF_DEST is greater than 1, the record represents a specific broadcast.

This field can be modified but caution is advised. The system has no way to determine if it is consistent with the number of destinations described in RTDANAMS.

RTRECTYP/RTDARTYP

A 3-byte character field containing the record type.

This field can be modified by the specific routing exit or the input record exit at OPEN time.

RTFUNCTN/RTDAFUNC

An 8-byte character field containing the function. This can be one of four conditions affecting the input record exit (open, put, close, and term-release.)

RTLRECL/RTDARECL

A fullword binary field specifying the logical record length for the print file

This field can be modified by the specific routing exit or the input record exit at OPEN time.

RTBLKSZ/RTDABLKS

A fullword binary field specifying the block size for the print file.

This field can be modified by the specific routing exit or the input record exit at OPEN time.

RTBUFSZ/RTDABUFS

A fullword binary field specifying the buffer size for the print file.

This field can be modified by the specific routing exit or the input record exit at OPEN time.

RTDAOPTN

A 16-byte character field containing the name of the Options record for this print job.

This field should not be modified.

RTDAOPTL

A halfword binary field containing the length of the options data used for this routing. This length must be 255 or less.

This field can be modified by either exit.

RTDAOPTD

A 256-byte character field containing up to 255 bytes of option data for this print job (the right-most byte of this field is ignored). They correspond to the options used for LPR. See the file description section for values.

This field can be modified by either exit.

RTDANAMS

A variable area containing a destination data area for each destination. The format of the destination data areas is as follows:

Field Name

Field Description

RTDANAML

A 2-byte binary field specifying the length of INET_NAME

RTDAPRTL

A 2-byte binary field specifying the length of PRT_NAME

RTDAINAM

A variable length character field up to 255 characters in length specifying the internet name or address of the destination host for the first or only destination. This field is case-sensitive.

RTDAPNAM

A variable length character field up to 255 characters in length specifying the name of the printer to be used at the remote host for the first or only destination. This field is case-sensitive.

These fields might be modified by either exit. Care should be taken to make sure that the resulting modifications are consistent. Also, if the routing data area was built by NPF, there will be no additional storage at the end for additions.

DSECT for routing data area

The DSECT for the routing data area is generated using the EZAPPFCD TYPE=DSECT macro.

```
*---------------------------------------------------------------------*
*                DSECT for Routing Data Area                          *
*---------------------------------------------------------------------*
RTDATA   DSECT                    DSECT to Describe Routing Data
RTDAMAJK DS    CL8                Major Route Key                      X
                                  JES -  Class of Output               X
                                  VTAM - LU Name
RTDAMINK DS    CL8                Minor Route Key                      X
                                  JES -  Destination of Output         X
                                  VTAM - nonblank
RTDASREX DS    CL8                Name of specific routing exit
RTDAIREX DS    CL8                Name of input record exit
RTDARTRS DS    F                  Success retain time in YDDDHHMM
RTDARTRU DS    F                  Failure retain time in YDDDHHMM
RTDARTRR DS    F                  Retry time in YDDDHHMM
RTDADEST DS    H                  Number of destinations               X
                                  One destination = normal routing
RTDARTRL DS    H                  Number of Retry Attempts
RTDATYPE DS    CL1                Route Type
RTDATYPN EQU   C'N'               Normal Routing
RTDATYPS EQU   C'S'               Specific Broadcast
RTRECTYP DS    CL3                Record type for print file
RTFUNCTN DS    CL8                The function in progress
RTLRECL  DS    F                  Logical record length for print file
RTBLKSZ  DS    F                  Block size for print file
RTBUFSZ  DS    F                  Size of I/O buffer
RTDAOPTN DS    CL16               Name of options record for this route
RTDAOPTL DS    H                  Length of the options data
RTDAOPTD DS    CL256              Options data for this route
RTDANAMS DS    0H                 Beginning of names
RTDADESD DSECT                    DSECT for names - This DSECT will be X
                                  repeated for each destination        X
                                  supported in this route = RTDADEST
RTDANAML DS    H                  Length of internet name for this     X
                                  destination
RTDAPRTL DS    H                  Length of printer name for this      X
                                  destination
RTDAINAM DS    CL256              Internet name for this destination
RTDAPNAM DS    CL256              Printer name for this destination
RTDADESE EQU   *
RTDADESL EQU   RTDADESE-RTDADESD
```

Figure 42. DSECT for routing data area

Appendix C. Sample NPF FSS writer definitions

```
***********************************************************************
*   The following samples show a breakdown of the FSSDEF definition
*   statement and PRINTER definition statement required to be placed
*   in the JES initialization job.
*   The samples are NOT executable code but rather a grouping of the
*   definitions for the following JES versions:
*
*                       JES2 V4 or V5
*                       JES3 V4
*
***********************************************************************
*            WARNING : This sample is NOT executable code.
***********************************************************************
*            COPYRIGHT = NONE.
***********************************************************************
```

Definitions for JES2 Version 4 and Version 5 are the same for FSS and PRT for the purposes of these examples.

```
*************************
*                       *
*   JES2 V4 or V5       *
*                       *
*************************

**********************************************
*           FSS WRITER DEFINITIONS           *
*                                            *
*  TCPFSS IS FOR PRINTER 5 AND PRINTER 6     *
*  TCPFS1 IS FOR PRINTER 7                   *
**********************************************
FSSDEF(TCPFSS) PROC=FSWTR,HASPFSSM=HASPFSSM,AUTOSTOP=YES
FSSDEF(TCPFS1) PROC=FSWTR,HASPFSSM=HASPFSSM,AUTOSTOP=YES

**********************************************
*          FSS PRINTER DEFINITIONS           *
**********************************************
PRT5    MODE=FSS,FSS=TCPFSS,DRAIN,CLASS=C
PRT6    MODE=FSS,FSS=TCPFSS,DRAIN,CLASS=D
PRT7    MODE=FSS,FSS=TCPFS1,DRAIN,CLASS=E
```

Figure 43. FSS writer definition sample for JES2 version 4 and 5

```
***************
*   JES3 V4   *
***************

**********************************************
*           FSS WRITER DEFINITIONS           *
*           WITH PRINTER (DEVICE)            *
*           DEFINITIONS FOLLOWING            *
*                                            *
*  FSSPRT1 IS FOR PRINTER 501                *
*  FSSPRT2 IS FOR PRINTER 502                *
*  FSSPRT3 IS FOR PRINTER 503                *
*  FSSPRT4 IS FOR PRINTER 504                *
**********************************************

FSSDEF,TYPE=WTR,FSSNAME=FSSPRT1,MSGDEST=JES,                          X
       PNAME=FSWTR,SYSTEM=MVS3,TERM=YES
DEVICE,DTYPE=PRT3820,JNAME=PRT501,MODE=FSS,                           X
       FSSNAME=FSSPRT1,JUNIT=(,MVS3,S1,ON),SETUPMSG=NO
FSSDEF,TYPE=WTR,FSSNAME=FSSPRT2,MSGDEST=JES,                          X
       PNAME=FSWTR,SYSTEM=MVS3,TERM=YES
DEVICE,DTYPE=PRT3820,JNAME=PRT502,MODE=FSS,                           X
       FSSNAME=FSSPRT2,JUNIT=(,MVS3,S1,ON),SETUPMSG=NO
FSSDEF,TYPE=WTR,FSSNAME=FSSPRT3,MSGDEST=JES,                          X
       PNAME=FSWTR,SYSTEM=MVS3,TERM=YES
DEVICE,DTYPE=PRT3820,JNAME=PRT503,MODE=FSS,                           X
       FSSNAME=FSSPRT3,JUNIT=(,MVS3,S1,ON),SETUPMSG=NO
FSSDEF,TYPE=WTR,FSSNAME=FSSPRT4,MSGDEST=JES,                          X
       PNAME=FSWTR,SYSTEM=MVS3,TERM=YES
DEVICE,DTYPE=PRT3820,JNAME=PRT504,MODE=FSS,                           X
       FSSNAME=FSSPRT4,JUNIT=(,MVS3,S1,ON),SETUPMSG=NO
```

Figure 44. FSS writer definition sample for JES3 version 4

Appendix D. The SCS printer emulator

The SCS (LU type 1) printer emulator transforms the SNA Character Stream into a VBA (variable blocked with ASA carriage control) data set. The following table describes the code points that are undefined, unsupported, supported with defaults or fully supported. Undefined and unsupported code points are rejected with an SNA sense code of function error (X'10030000'). Supported code points with invalid parameters or without all parameters available in the same chain are rejected with an SNA sense code of parameter error (X'10050000').

Table 18. SCS code points

	x0	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**	**x7**	**x8**	**x9**	**xA**	**xB**	**xC**	**xD**	**xE**	**xF**
0x	Null	(1)	(1)	(1)	SEL (9)	HT	RNL (4)	(1)	GE (2)	SPS (3)	RPT (3)	VT	FF	CR	SO	SI
1x	(1)	DC1 (3)	DC2 (3)	DC3 (3)	ENP (3)	NL	BS	POC (3)	(1)	(1)	UBS (3)	CU1 (3)	IFS (4)	IGS (4)	IRS (4)	IUS (7)
2x	(1)	(1)	(1)	WUS	INP (3)	LF	(1)	(1)	SA (2)	(1)	SW (3)	CSP (10)	(1)	(1)	(1)	BEL (3)
3x	(1)	(1)	SYN (3)	IR (4)	PP	TRN (2)	EBS (6)	(1)	SBS (3)	IT (3)	RFF (5)	CU3 (3)	DC4 (3)	(1)	(1)	SUB
4x		RSP (7)														
5x																
6x																
7x																
8x																
9x																
Ax																
Bx																
Cx											SHY (8)					
Dx																

Table 18. SCS code points (continued)

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
Ex		ESP (7)														
Fx																

Notes on the table:

1. Undefined code point - function error.
2. Unsupported code point - function error.
3. Defaults to no operation - function ignored.
4. Defaults to new line (NL).
5. Defaults to form feed (FF).
6. Defaults to backspace (BS).
7. Defaults to space (X'40').
8. Defaults to dash (X'60').
9. Vertical channel select is supported. Select left/right platten is ignored. Select magnetic stripe reader/writer is unsupported.
10. Set Horizontal Format and Set Vertical Format are supported. Start of Format is ignored if at left margin, defaults to new line (NL) if not at left margin. Set Line Density, Set Graphic Escape Action, Set Chain Image and Set Print Density are ignored.

Double byte character strings (DBCS) begin with the shift_out (SO) code point and end with the shift_in (SI) code point. Valid characters between the SI and SO are X'4040' and any pair of bytes each in the range X'41' to X'FE'. Illegal characters found in a double byte character string or double byte character strings that do not complete in the same chain are rejected with an SNA sense code of data error (X'10010000').

All unspecified code points are passed to LPR as EBCDIC data. See the z/OS Communications Server: IP User's Guide and Commands to determine the EBCDIC-to-ASCII translation performed by LPR. See *SNA - Sessions Between Logical Units* for more information about SCS data streams.

Appendix E. Related protocol specifications

This appendix lists the related protocol specifications (RFCs) for TCP/IP. The Internet Protocol suite is still evolving through requests for comments (RFC). New protocols are being designed and implemented by researchers and are brought to the attention of the Internet community in the form of RFCs. Some of these protocols are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement these particular functions or protocols. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

RFCs are available at http://www.rfc-editor.org/rfc.html.

Draft RFCs that have been implemented in this and previous Communications Server releases are listed at the end of this topic.

Many features of TCP/IP Services are based on the following RFCs:

RFC
: **Title and Author**

RFC 652
: *Telnet output carriage-return disposition option* D. Crocker

RFC 653
: *Telnet output horizontal tabstops option* D. Crocker

RFC 654
: *Telnet output horizontal tab disposition option* D. Crocker

RFC 655
: *Telnet output formfeed disposition option* D. Crocker

RFC 657
: *Telnet output vertical tab disposition option* D. Crocker

RFC 658
: *Telnet output linefeed disposition* D. Crocker

RFC 698
: *Telnet extended ASCII option* T. Mock

RFC 726
: *Remote Controlled Transmission and Echoing Telnet option* J. Postel, D. Crocker

RFC 727
: *Telnet logout option* M.R. Crispin

RFC 732
: *Telnet Data Entry Terminal option* J.D. Day

RFC 733
: *Standard for the format of ARPA network text messages* D. Crocker, J. Vittal, K.T. Pogran, D.A. Henderson

RFC 734
: *SUPDUP Protocol* M.R. Crispin

RFC 735
: *Revised Telnet byte macro option* D. Crocker, R.H. Gumpertz

RFC 736
: *Telnet SUPDUP option* M.R. Crispin

RFC 749
: *Telnet SUPDUP—Output option* B. Greenberg

RFC 765
: *File Transfer Protocol specification* J. Postel

RFC 768
User Datagram Protocol J. Postel

RFC 779
Telnet send-location option E. Killian

RFC 791
Internet Protocol J. Postel

RFC 792
Internet Control Message Protocol J. Postel

RFC 793
Transmission Control Protocol J. Postel

RFC 820
Assigned numbers J. Postel

RFC 823
DARPA Internet gateway R. Hinden, A. Sheltzer

RFC 826
Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware D. Plummer

RFC 854
Telnet Protocol Specification J. Postel, J. Reynolds

RFC 855
Telnet Option Specification J. Postel, J. Reynolds

RFC 856
Telnet Binary Transmission J. Postel, J. Reynolds

RFC 857
Telnet Echo Option J. Postel, J. Reynolds

RFC 858
Telnet Suppress Go Ahead Option J. Postel, J. Reynolds

RFC 859
Telnet Status Option J. Postel, J. Reynolds

RFC 860
Telnet Timing Mark Option J. Postel, J. Reynolds

RFC 861
Telnet Extended Options: List Option J. Postel, J. Reynolds

RFC 862
Echo Protocol J. Postel

RFC 863
Discard Protocol J. Postel

RFC 864
Character Generator Protocol J. Postel

RFC 865
Quote of the Day Protocol J. Postel

RFC 868
Time Protocol J. Postel, K. Harrenstien

RFC 877
Standard for the transmission of IP datagrams over public data networks J.T. Korb

RFC 883
Domain names: Implementation specification P.V. Mockapetris

RFC 884
Telnet terminal type option M. Solomon, E. Wimmers

RFC 885
Telnet end of record option J. Postel

RFC 894
Standard for the transmission of IP datagrams over Ethernet networks C. Hornig

RFC 896
Congestion control in IP/TCP internetworks J. Nagle

RFC 903
Reverse Address Resolution Protocol R. Finlayson, T. Mann, J. Mogul, M. Theimer

RFC 904
Exterior Gateway Protocol formal specification D. Mills

RFC 919
Broadcasting Internet Datagrams J. Mogul

RFC 922
Broadcasting Internet datagrams in the presence of subnets J. Mogul

RFC 927
TACACS user identification Telnet option B.A. Anderson

RFC 933
Output marking Telnet option S. Silverman

RFC 946
Telnet terminal location number option R. Nedved

RFC 950
Internet Standard Subnetting Procedure J. Mogul, J. Postel

RFC 952
DoD Internet host table specification K. Harrenstien, M. Stahl, E. Feinler

RFC 959
File Transfer Protocol J. Postel, J.K. Reynolds

RFC 961
Official ARPA-Internet protocols J.K. Reynolds, J. Postel

RFC 974
Mail routing and the domain system C. Partridge

RFC 1001
Protocol standard for a NetBIOS service on a TCP/UDP transport: Concepts and methods NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force

RFC 1002
Protocol Standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications NetBios Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, End-to-End Services Task Force

RFC 1006
ISO transport services on top of the TCP: Version 3 M.T. Rose, D.E. Cass

RFC 1009
Requirements for Internet gateways R. Braden, J. Postel

RFC 1011
Official Internet protocols J. Reynolds, J. Postel

RFC 1013
X Window System Protocol, version 11: Alpha update April 1987 R. Scheifler

RFC 1014
XDR: External Data Representation standard Sun Microsystems

RFC 1027
Using ARP to implement transparent subnet gateways S. Carl-Mitchell, J. Quarterman

RFC 1032
Domain administrators guide M. Stahl

RFC 1033
Domain administrators operations guide M. Lottor

RFC 1034
Domain names—concepts and facilities P.V. Mockapetris

RFC 1035
Domain names—implementation and specification P.V. Mockapetris

RFC 1038
Draft revised IP security option M. St. Johns

RFC 1041
Telnet 3270 regime option Y. Rekhter

RFC 1042
Standard for the transmission of IP datagrams over IEEE 802 networks J. Postel, J. Reynolds

RFC 1043
Telnet Data Entry Terminal option: DODIIS implementation A. Yasuda, T. Thompson

RFC 1044
Internet Protocol on Network System's HYPERchannel: Protocol specification K. Hardwick, J. Lekashman

RFC 1053
Telnet X.3 PAD option S. Levy, T. Jacobson

RFC 1055
Nonstandard for transmission of IP datagrams over serial lines: SLIP J. Romkey

RFC 1057
RPC: Remote Procedure Call Protocol Specification: Version 2 Sun Microsystems

RFC 1058
Routing Information Protocol C. Hedrick

RFC 1060
Assigned numbers J. Reynolds, J. Postel

RFC 1067
Simple Network Management Protocol J.D. Case, M. Fedor, M.L. Schoffstall, J. Davin

RFC 1071
Computing the Internet checksum R.T. Braden, D.A. Borman, C. Partridge

RFC 1072
TCP extensions for long-delay paths V. Jacobson, R.T. Braden

RFC 1073
Telnet window size option D. Waitzman

RFC 1079
Telnet terminal speed option C. Hedrick

RFC 1085
ISO presentation services on top of TCP/IP based internets M.T. Rose

RFC 1091
Telnet terminal-type option J. VanBokkelen

RFC 1094
NFS: Network File System Protocol specification Sun Microsystems

RFC 1096
Telnet X display location option G. Marcy

RFC 1101
DNS encoding of network names and other types P. Mockapetris

RFC 1112
Host extensions for IP multicasting S.E. Deering

RFC 1113
Privacy enhancement for Internet electronic mail: Part I — message encipherment and authentication procedures J. Linn

RFC 1118
Hitchhikers Guide to the Internet E. Krol

RFC 1122
Requirements for Internet Hosts—Communication Layers R. Braden, Ed.

RFC 1123
Requirements for Internet Hosts—Application and Support R. Braden, Ed.

RFC 1146
TCP alternate checksum options J. Zweig, C. Partridge

RFC 1155
Structure and identification of management information for TCP/IP-based internets M. Rose, K. McCloghrie

RFC 1156
Management Information Base for network management of TCP/IP-based internets K. McCloghrie, M. Rose

RFC 1157
Simple Network Management Protocol (SNMP) J. Case, M. Fedor, M. Schoffstall, J. Davin

RFC 1158
Management Information Base for network management of TCP/IP-based internets: MIB-II M. Rose

RFC 1166
Internet numbers S. Kirkpatrick, M.K. Stahl, M. Recker

RFC 1179
Line printer daemon protocol L. McLaughlin

RFC 1180
TCP/IP tutorial T. Socolofsky, C. Kale

RFC 1183
New DNS RR Definitions C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris

RFC 1184
Telnet Linemode Option D. Borman

RFC 1186
MD4 Message Digest Algorithm R.L. Rivest

RFC 1187
Bulk Table Retrieval with the SNMP M. Rose, K. McCloghrie, J. Davin

RFC 1188
Proposed Standard for the Transmission of IP Datagrams over FDDI Networks D. Katz

RFC 1190
Experimental Internet Stream Protocol: Version 2 (ST-II) C. Topolcic

RFC 1191
Path MTU discovery J. Mogul, S. Deering

RFC 1198
FYI on the X window system R. Scheifler

RFC 1207
FYI on Questions and Answers: Answers to commonly asked "experienced Internet user" questions G. Malkin, A. Marine, J. Reynolds

RFC 1208
Glossary of networking terms O. Jacobsen, D. Lynch

RFC 1213
Management Information Base for Network Management of TCP/IP-based internets: MIB-II K. McCloghrie, M.T. Rose

RFC 1215
Convention for defining traps for use with the SNMP M. Rose

RFC 1227
SNMP MUX protocol and MIB M.T. Rose

RFC 1228
SNMP-DPI: Simple Network Management Protocol Distributed Program Interface G. Carpenter, B. Wijnen

RFC 1229
Extensions to the generic-interface MIB K. McCloghrie

RFC 1230
IEEE 802.4 Token Bus MIB K. McCloghrie, R. Fox

RFC 1231
IEEE 802.5 Token Ring MIB K. McCloghrie, R. Fox, E. Decker

RFC 1236
IP to X.121 address mapping for DDN L. Morales, P. Hasse

RFC 1256
ICMP Router Discovery Messages S. Deering, Ed.

RFC 1267
Border Gateway Protocol 3 (BGP-3) K. Lougheed, Y. Rekhter

RFC 1268
Application of the Border Gateway Protocol in the Internet Y. Rekhter, P. Gross

RFC 1269
Definitions of Managed Objects for the Border Gateway Protocol: Version 3 S. Willis, J. Burruss

RFC 1270
SNMP Communications Services F. Kastenholz, ed.

RFC 1285
FDDI Management Information Base J. Case

RFC 1315
Management Information Base for Frame Relay DTEs C. Brown, F. Baker, C. Carvalho

RFC 1321
The MD5 Message-Digest Algorithm R. Rivest

RFC 1323
TCP Extensions for High Performance V. Jacobson, R. Braden, D. Borman

RFC 1325
FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions G. Malkin, A. Marine

RFC 1327
Mapping between X.400 (1988)/ISO 10021 and RFC 822 S. Hardcastle-Kille

RFC 1340
Assigned Numbers J. Reynolds, J. Postel

RFC 1344
Implications of MIME for Internet Mail Gateways N. Bornstein

RFC 1349
Type of Service in the Internet Protocol Suite P. Almquist

RFC 1351
SNMP Administrative Model J. Davin, J. Galvin, K. McCloghrie

RFC 1352
SNMP Security Protocols J. Galvin, K. McCloghrie, J. Davin

RFC 1353
Definitions of Managed Objects for Administration of SNMP Parties K. McCloghrie, J. Davin, J. Galvin

RFC 1354
IP Forwarding Table MIB F. Baker

RFC 1356
Multiprotocol Interconnect® on X.25 and ISDN in the Packet Mode A. Malis, D. Robinson, R. Ullmann

RFC 1358
Charter of the Internet Architecture Board (IAB) L. Chapin

RFC 1363
A Proposed Flow Specification C. Partridge

RFC 1368
Definition of Managed Objects for IEEE 802.3 Repeater Devices D. McMaster, K. McCloghrie

RFC 1372
Telnet Remote Flow Control Option C. L. Hedrick, D. Borman

RFC 1374
IP and ARP on HIPPI J. Renwick, A. Nicholson

RFC 1381
SNMP MIB Extension for X.25 LAPB D. Throop, F. Baker

RFC 1382
SNMP MIB Extension for the X.25 Packet Layer D. Throop

RFC 1387
RIP Version 2 Protocol Analysis G. Malkin

RFC 1388
RIP Version 2 Carrying Additional Information G. Malkin

RFC 1389
RIP Version 2 MIB Extensions G. Malkin, F. Baker

RFC 1390
Transmission of IP and ARP over FDDI Networks D. Katz

RFC 1393
Traceroute Using an IP Option G. Malkin

RFC 1398
Definitions of Managed Objects for the Ethernet-Like Interface Types F. Kastenholz

RFC 1408
Telnet Environment Option D. Borman, Ed.

RFC 1413
Identification Protocol M. St. Johns

RFC 1416
Telnet Authentication Option D. Borman, ed.

RFC 1420
SNMP over IPX S. Bostock

RFC 1428
Transition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME G. Vaudreuil

RFC 1442
Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1443
Textual Conventions for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1445
Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2) J. Galvin, K. McCloghrie

RFC 1447
Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2) K. McCloghrie, J. Galvin

RFC 1448
Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1464
Using the Domain Name System to Store Arbitrary String Attributes R. Rosenbaum

RFC 1469
IP Multicast over Token-Ring Local Area Networks T. Pusateri

RFC 1483
Multiprotocol Encapsulation over ATM Adaptation Layer 5 Juha Heinanen

RFC 1514
Host Resources MIB P. Grillo, S. Waldbusser

RFC 1516
Definitions of Managed Objects for IEEE 802.3 Repeater Devices D. McMaster, K. McCloghrie

RFC 1521
MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies N. Borenstein, N. Freed

RFC 1535
A Security Problem and Proposed Correction With Widely Deployed DNS Software E. Gavron

RFC 1536
Common DNS Implementation Errors and Suggested Fixes A. Kumar, J. Postel, C. Neuman, P. Danzig, S. Miller

RFC 1537
Common DNS Data File Configuration Errors P. Beertema

RFC 1540
Internet Official Protocol Standards J. Postel

RFC 1571
Telnet Environment Option Interoperability Issues D. Borman

RFC 1572
Telnet Environment Option S. Alexander

RFC 1573
Evolution of the Interfaces Group of MIB-II K. McCloghrie, F. Kastenholz

RFC 1577
Classical IP and ARP over ATM M. Laubach

RFC 1583
OSPF Version 2 J. Moy

RFC 1591
Domain Name System Structure and Delegation J. Postel

RFC 1592
Simple Network Management Protocol Distributed Protocol Interface Version 2.0 B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

RFC 1594
FYI on Questions and Answers— Answers to Commonly Asked "New Internet User" Questions A. Marine, J. Reynolds, G. Malkin

RFC 1644
T/TCP — TCP Extensions for Transactions Functional Specification R. Braden

RFC 1646
TN3270 Extensions for LUname and Printer Selection C. Graves, T. Butts, M. Angel

RFC 1647
TN3270 Enhancements B. Kelly

RFC 1652
SMTP Service Extension for 8bit-MIMEtransport J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker

RFC 1664
Using the Internet DNS to Distribute RFC1327 Mail Address Mapping Tables C. Allochio, A. Bonito, B. Cole, S. Giordano, R. Hagens

RFC 1693
An Extension to TCP: Partial Order Service T. Connolly, P. Amer, P. Conrad

RFC 1695
Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2 M. Ahmed, K. Tesink

RFC 1701
Generic Routing Encapsulation (GRE) S. Hanks, T. Li, D. Farinacci, P. Traina

RFC 1702
Generic Routing Encapsulation over IPv4 networks S. Hanks, T. Li, D. Farinacci, P. Traina

RFC 1706
DNS NSAP Resource Records B. Manning, R. Colella

RFC 1712
DNS Encoding of Geographical Location C. Farrell, M. Schulze, S. Pleitner D. Baldoni

RFC 1713
Tools for DNS debugging A. Romao

RFC 1723
RIP Version 2—Carrying Additional Information G. Malkin

RFC 1752
The Recommendation for the IP Next Generation Protocol S. Bradner, A. Mankin

RFC 1766
Tags for the Identification of Languages H. Alvestrand

RFC 1771
A Border Gateway Protocol 4 (BGP-4) Y. Rekhter, T. Li

RFC 1794
DNS Support for Load Balancing T. Brisco

RFC 1819
Internet Stream Protocol Version 2 (ST2) Protocol Specification—Version ST2+ L. Delgrossi, L. Berger Eds.

RFC 1826
IP Authentication Header R. Atkinson

RFC 1828
IP Authentication using Keyed MD5 P. Metzger, W. Simpson

RFC 1829
The ESP DES-CBC Transform P. Karn, P. Metzger, W. Simpson

RFC 1830
SMTP Service Extensions for Transmission of Large and Binary MIME Messages G. Vaudreuil

RFC 1831
RPC: Remote Procedure Call Protocol Specification Version 2 R. Srinivasan

RFC 1832
XDR: External Data Representation Standard R. Srinivasan

RFC 1833
Binding Protocols for ONC RPC Version 2 R. Srinivasan

RFC 1850
OSPF Version 2 Management Information Base F. Baker, R. Coltun

RFC 1854
SMTP Service Extension for Command Pipelining N. Freed

RFC 1869
SMTP Service Extensions J. Klensin, N. Freed, M. Rose, E. Stefferud, D. Crocker

RFC 1870
SMTP Service Extension for Message Size Declaration J. Klensin, N. Freed, K. Moore

RFC 1876
A Means for Expressing Location Information in the Domain Name System C. Davis, P. Vixie, T. Goodwin, I. Dickinson

RFC 1883
Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden

RFC 1884
IP Version 6 Addressing Architecture R. Hinden, S. Deering, Eds.

RFC 1886
DNS Extensions to support IP version 6 S. Thomson, C. Huitema

RFC 1888
OSI NSAPs and IPv6 J. Bound, B. Carpenter, D. Harrington, J. Houldsworth, A. Lloyd

RFC 1891
SMTP Service Extension for Delivery Status Notifications K. Moore

RFC 1892
The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages G. Vaudreuil

RFC 1894
*An Extensible Message Format for Delivery Status Notifications*K. Moore, G. Vaudreuil

RFC 1901
Introduction to Community-based SNMPv2 J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1902
Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1903
Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1904
Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1905
Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1906
Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1907
Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2) J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1908
Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 1912
Common DNS Operational and Configuration Errors D. Barr

RFC 1918
Address Allocation for Private Internets Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, E. Lear

RFC 1928
SOCKS Protocol Version 5 M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones

RFC 1930
Guidelines for creation, selection, and registration of an Autonomous System (AS) J. Hawkinson, T. Bates

RFC 1939
Post Office Protocol-Version 3 J. Myers, M. Rose

RFC 1981
Path MTU Discovery for IP version 6 J. McCann, S. Deering, J. Mogul

RFC 1982
Serial Number Arithmetic R. Elz, R. Bush

RFC 1985
SMTP Service Extension for Remote Message Queue Starting J. De Winter

RFC 1995
Incremental Zone Transfer in DNS M. Ohta

RFC 1996
A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY) P. Vixie

RFC 2010
Operational Criteria for Root Name Servers B. Manning, P. Vixie

RFC 2011
SNMPv2 Management Information Base for the Internet Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2012
SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2013
SNMPv2 Management Information Base for the User Datagram Protocol using SMIv2 K. McCloghrie, Ed.

RFC 2018
TCP Selective Acknowledgement Options M. Mathis, J. Mahdavi, S. Floyd, A. Romanow

RFC 2026
The Internet Standards Process — Revision 3 S. Bradner

RFC 2030
Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI D. Mills

RFC 2033
Local Mail Transfer Protocol J. Myers

RFC 2034
*SMTP Service Extension for Returning Enhanced Error Codes*N. Freed

RFC 2040
*The RC5, RC5–CBC, RC-5–CBC-Pad, and RC5–CTS Algorithms*R. Baldwin, R. Rivest

RFC 2045
Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies N. Freed, N. Borenstein

RFC 2052
A DNS RR for specifying the location of services (DNS SRV) A. Gulbrandsen, P. Vixie

RFC 2065
Domain Name System Security Extensions D. Eastlake 3rd, C. Kaufman

RFC 2066
TELNET CHARSET Option R. Gellens

RFC 2080
RIPng for IPv6 G. Malkin, R. Minnear

RFC 2096
IP Forwarding Table MIB F. Baker

RFC 2104
HMAC: Keyed-Hashing for Message Authentication H. Krawczyk, M. Bellare, R. Canetti

RFC 2119
Keywords for use in RFCs to Indicate Requirement Levels S. Bradner

RFC 2133
Basic Socket Interface Extensions for IPv6 R. Gilligan, S. Thomson, J. Bound, W. Stevens

RFC 2136
Dynamic Updates in the Domain Name System (DNS UPDATE) P. Vixie, Ed., S. Thomson, Y. Rekhter, J. Bound

RFC 2137
Secure Domain Name System Dynamic Update D. Eastlake 3rd

RFC 2163
Using the Internet DNS to Distribute MIXER Conformant Global Address Mapping (MCGAM) C. Allocchio

RFC 2168
Resolution of Uniform Resource Identifiers using the Domain Name System R. Daniel, M. Mealling

RFC 2178
OSPF Version 2 J. Moy

RFC 2181
Clarifications to the DNS Specification R. Elz, R. Bush

RFC 2205
Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin

RFC 2210
The Use of RSVP with IETF Integrated Services J. Wroclawski

RFC 2211
Specification of the Controlled-Load Network Element Service J. Wroclawski

RFC 2212
Specification of Guaranteed Quality of Service S. Shenker, C. Partridge, R. Guerin

RFC 2215
General Characterization Parameters for Integrated Service Network Elements S. Shenker, J. Wroclawski

RFC 2217
Telnet Com Port Control Option G. Clarke

RFC 2219
Use of DNS Aliases for Network Services M. Hamilton, R. Wright

RFC 2228
FTP Security Extensions M. Horowitz, S. Lunt

RFC 2230
Key Exchange Delegation Record for the DNS R. Atkinson

RFC 2233
The Interfaces Group MIB using SMIv2 K. McCloghrie, F. Kastenholz

RFC 2240
A Legal Basis for Domain Name Allocation O. Vaughn

RFC 2246
The TLS Protocol Version 1.0 T. Dierks, C. Allen

RFC 2251
Lightweight Directory Access Protocol (v3) M. Wahl, T. Howes, S. Kille

RFC 2253
Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names M. Wahl, S. Kille, T. Howes

RFC 2254
The String Representation of LDAP Search Filters T. Howes

RFC 2261
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 2262
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 2271
An Architecture for Describing SNMP Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 2273
SNMPv3 Applications D. Levi, P. Meyer, B. Stewartz

RFC 2274
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 2275
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 2279
UTF-8, a transformation format of ISO 10646 F. Yergeau

RFC 2292
Advanced Sockets API for IPv6 W. Stevens, M. Thomas

RFC 2308
Negative Caching of DNS Queries (DNS NCACHE) M. Andrews

RFC 2317
Classless IN-ADDR.ARPA delegation H. Eidnes, G. de Groot, P. Vixie

RFC 2320
Definitions of Managed Objects for Classical IP and ARP Over ATM Using SMIv2 (IPOA-MIB) M. Greene, J. Luciani, K. White, T. Kuo

RFC 2328
OSPF Version 2 J. Moy

RFC 2345
Domain Names and Company Name Retrieval J. Klensin, T. Wolf, G. Oglesby

RFC 2352
A Convention for Using Legal Names as Domain Names O. Vaughn

RFC 2355
TN3270 Enhancements B. Kelly

RFC 2358
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson

RFC 2373
IP Version 6 Addressing Architecture R. Hinden, S. Deering

RFC 2374
An IPv6 Aggregatable Global Unicast Address Format R. Hinden, M. O'Dell, S. Deering

RFC 2375
IPv6 Multicast Address Assignments R. Hinden, S. Deering

RFC 2385
Protection of BGP Sessions via the TCP MD5 Signature Option A. Hefferman

RFC 2389
Feature negotiation mechanism for the File Transfer Protocol P. Hethmon, R. Elz

RFC 2401
Security Architecture for Internet Protocol S. Kent, R. Atkinson

RFC 2402
IP Authentication Header S. Kent, R. Atkinson

RFC 2403
The Use of HMAC-MD5–96 within ESP and AH C. Madson, R. Glenn

RFC 2404
The Use of HMAC-SHA–1–96 within ESP and AH C. Madson, R. Glenn

RFC 2405
The ESP DES-CBC Cipher Algorithm With Explicit IV C. Madson, N. Doraswamy

RFC 2406
IP Encapsulating Security Payload (ESP) S. Kent, R. Atkinson

RFC 2407
*The Internet IP Security Domain of Interpretation for ISAKMP*D. Piper

RFC 2408
Internet Security Association and Key Management Protocol (ISAKMP) D. Maughan, M. Schertler, M. Schneider, J. Turner

RFC 2409
The Internet Key Exchange (IKE) D. Harkins, D. Carrel

RFC 2410
The NULL Encryption Algorithm and Its Use With IPsec R. Glenn, S. Kent,

RFC 2428
FTP Extensions for IPv6 and NATs M. Allman, S. Ostermann, C. Metz

RFC 2445
Internet Calendaring and Scheduling Core Object Specification (iCalendar) F. Dawson, D. Stenerson

RFC 2459
Internet X.509 Public Key Infrastructure Certificate and CRL Profile R. Housley, W. Ford, W. Polk, D. Solo

RFC 2460
Internet Protocol, Version 6 (IPv6) Specification S. Deering, R. Hinden

RFC 2461
Neighbor Discovery for IP Version 6 (IPv6) T. Narten, E. Nordmark, W. Simpson

RFC 2462
IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten

RFC 2463
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

RFC 2464
Transmission of IPv6 Packets over Ethernet Networks M. Crawford

RFC 2466
Management Information Base for IP Version 6: ICMPv6 Group D. Haskin, S. Onishi

RFC 2476
Message Submission R. Gellens, J. Klensin

RFC 2487
SMTP Service Extension for Secure SMTP over TLS P. Hoffman

RFC 2505
Anti-Spam Recommendations for SMTP MTAs G. Lindberg

RFC 2523
Photuris: Extended Schemes and Attributes P. Karn, W. Simpson

RFC 2535
Domain Name System Security Extensions D. Eastlake 3rd

RFC 2538
Storing Certificates in the Domain Name System (DNS) D. Eastlake 3rd, O. Gudmundsson

RFC 2539
Storage of Diffie-Hellman Keys in the Domain Name System (DNS) D. Eastlake 3rd

RFC 2540
Detached Domain Name System (DNS) Information D. Eastlake 3rd

RFC 2554
SMTP Service Extension for Authentication J. Myers

RFC 2570
Introduction to Version 3 of the Internet-standard Network Management Framework J. Case, R. Mundy, D. Partain, B. Stewart

RFC 2571
An Architecture for Describing SNMP Management Frameworks B. Wijnen, D. Harrington, R. Presuhn

RFC 2572
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 2573
SNMP Applications D. Levi, P. Meyer, B. Stewart

RFC 2574
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 2575
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 2576
Co-Existence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework R. Frye, D. Levi, S. Routhier, B. Wijnen

RFC 2578
Structure of Management Information Version 2 (SMIv2) K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2579
Textual Conventions for SMIv2 K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2580
Conformance Statements for SMIv2 K. McCloghrie, D. Perkins, J. Schoenwaelder

RFC 2581
TCP Congestion Control M. Allman, V. Paxson, W. Stevens

RFC 2583
Guidelines for Next Hop Client (NHC) Developers R. Carlson, L. Winkler

RFC 2591
Definitions of Managed Objects for Scheduling Management Operations D. Levi, J. Schoenwaelder

RFC 2625
IP and ARP over Fibre Channel M. Rajagopal, R. Bhagwat, W. Rickard

RFC 2635
Don't SPEW A Set of Guidelines for Mass Unsolicited Mailings and Postings (spam)* S. Hambridge, A. Lunde

RFC 2637
Point-to-Point Tunneling Protocol K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, G. Zorn

RFC 2640
Internationalization of the File Transfer Protocol B. Curtin

RFC 2665
Definitions of Managed Objects for the Ethernet-like Interface Types J. Flick, J. Johnson

RFC 2671
Extension Mechanisms for DNS (EDNS0) P. Vixie

RFC 2672
Non-Terminal DNS Name Redirection M. Crawford

RFC 2675
IPv6 Jumbograms D. Borman, S. Deering, R. Hinden

RFC 2710
Multicast Listener Discovery (MLD) for IPv6 S. Deering, W. Fenner, B. Haberman

RFC 2711
IPv6 Router Alert Option C. Partridge, A. Jackson

RFC 2740
OSPF for IPv6 R. Coltun, D. Ferguson, J. Moy

RFC 2753
A Framework for Policy-based Admission Control R. Yavatkar, D. Pendarakis, R. Guerin

RFC 2782
A DNS RR for specifying the location of services (DNS SRV) A. Gubrandsen, P. Vixix, L. Esibov

RFC 2821
Simple Mail Transfer Protocol J. Klensin, Ed.

RFC 2822
Internet Message Format P. Resnick, Ed.

RFC 2840
TELNET KERMIT OPTION J. Altman, F. da Cruz

RFC 2845
Secret Key Transaction Authentication for DNS (TSIG) P. Vixie, O. Gudmundsson, D. Eastlake 3rd, B. Wellington

RFC 2851
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 2852
Deliver By SMTP Service Extension D. Newman

RFC 2874
DNS Extensions to Support IPv6 Address Aggregation and Renumbering M. Crawford, C. Huitema

RFC 2915
The Naming Authority Pointer (NAPTR) DNS Resource Record M. Mealling, R. Daniel

RFC 2920
SMTP Service Extension for Command Pipelining N. Freed

RFC 2930
Secret Key Establishment for DNS (TKEY RR) D. Eastlake, 3rd

RFC 2941
Telnet Authentication Option T. Ts'o, ed., J. Altman

RFC 2942
Telnet Authentication: Kerberos Version 5 T. Ts'o

RFC 2946
Telnet Data Encryption Option T. Ts'o

RFC 2952
Telnet Encryption: DES 64 bit Cipher Feedback T. Ts'o

RFC 2953
Telnet Encryption: DES 64 bit Output Feedback T. Ts'o

RFC 2992
Analysis of an Equal-Cost Multi-Path Algorithm C. Hopps

RFC 3019
IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol B. Haberman, R. Worzella

RFC 3060
Policy Core Information Model—Version 1 Specification B. Moore, E. Ellesson, J. Strassner, A. Westerinen

RFC 3152
Delegation of IP6.ARPA R. Bush

RFC 3164
The BSD Syslog Protocol C. Lonvick

RFC 3207
SMTP Service Extension for Secure SMTP over Transport Layer Security P. Hoffman

RFC 3226
DNSSEC and IPv6 A6 aware server/resolver message size requirements O. Gudmundsson

RFC 3291
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 3363
Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System R. Bush, A. Durand, B. Fink, O. Gudmundsson, T. Hain

RFC 3376
Internet Group Management Protocol, Version 3 B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan

RFC 3390
Increasing TCP's Initial Window M. Allman, S. Floyd, C. Partridge

RFC 3410
Introduction and Applicability Statements for Internet-Standard Management Framework J. Case, R. Mundy, D. Partain, B. Stewart

RFC 3411
An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks D. Harrington, R. Presuhn, B. Wijnen

RFC 3412
Message Processing and Dispatching for the Simple Network Management Protocol (SNMP) J. Case, D. Harrington, R. Presuhn, B. Wijnen

RFC 3413
Simple Network Management Protocol (SNMP) Applications D. Levi, P. Meyer, B. Stewart

RFC 3414
User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) U. Blumenthal, B. Wijnen

RFC 3415
View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP) B. Wijnen, R. Presuhn, K. McCloghrie

RFC 3416
Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP) R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3417
Transport Mappings for the Simple Network Management Protocol (SNMP) R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3418
Management Information Base (MIB) for the Simple Network Management Protocol (SNMP) R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser

RFC 3419
Textual Conventions for Transport Addresses M. Daniele, J. Schoenwaelder

RFC 3484
Default Address Selection for Internet Protocol version 6 (IPv6) R. Draves

RFC 3493
Basic Socket Interface Extensions for IPv6 R. Gilligan, S. Thomson, J. Bound, J. McCann, W. Stevens

RFC 3513
Internet Protocol Version 6 (IPv6) Addressing Architecture R. Hinden, S. Deering

RFC 3526
More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE) T. Kivinen, M. Kojo

RFC 3542
Advanced Sockets Application Programming Interface (API) for IPv6 W. Richard Stevens, M. Thomas, E. Nordmark, T. Jinmei

RFC 3566
The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec S. Frankel, H. Herbert

RFC 3569
An Overview of Source-Specific Multicast (SSM) S. Bhattacharyya, Ed.

RFC 3584
Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework R. Frye, D. Levi, S. Routhier, B. Wijnen

RFC 3602
The AES-CBC Cipher Algorithm and Its Use with IPsec S. Frankel, R. Glenn, S. Kelly

RFC 3629
UTF-8, a transformation format of ISO 10646 R. Kermode, C. Vicisano

RFC 3658
Delegation Signer (DS) Resource Record (RR) O. Gudmundsson

RFC 3678
Socket Interface Extensions for Multicast Source Filters D. Thaler, B. Fenner, B. Quinn

RFC 3715
IPsec-Network Address Translation (NAT) Compatibility Requirements B. Aboba, W. Dixon

RFC 3810
Multicast Listener Discovery Version 2 (MLDv2) for IPv6 R. Vida, Ed., L. Costa, Ed.

RFC 3826
The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model U. Blumenthal, F. Maino, K McCloghrie.

RFC 3947
Negotiation of NAT-Traversal in the IKE T. Kivinen, B. Swander, A. Huttunen, V. Volpe

RFC 3948
UDP Encapsulation of IPsec ESP Packets A. Huttunen, B. Swander, V. Volpe, L. DiBurro, M. Stenberg

RFC 4001
Textual Conventions for Internet Network Addresses M. Daniele, B. Haberman, S. Routhier, J. Schoenwaelder

RFC 4007
IPv6 Scoped Address Architecture S. Deering, B. Haberman, T. Jinmei, E. Nordmark, B. Zill

RFC 4022
Management Information Base for the Transmission Control Protocol (TCP) R. Raghunarayan

RFC 4106
The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP) J. Viega, D. McGrew

RFC 4109
Algorithms for Internet Key Exchange version 1 (IKEv1) P. Hoffman

RFC 4113
Management Information Base for the User Datagram Protocol (UDP) B. Fenner, J. Flick

RFC 4191
Default Router Preferences and More-Specific Routes R. Draves, D. Thaler

RFC 4217
Securing FTP with TLS P. Ford-Hutchinson

RFC 4292
IP Forwarding Table MIB B. Haberman

RFC 4293
Management Information Base for the Internet Protocol (IP) S. Routhier

RFC 4301
Security Architecture for the Internet Protocol S. Kent, K. Seo

RFC 4302
IP Authentication Header S. Kent

RFC 4303
IP Encapsulating Security Payload (ESP) S. Kent

RFC 4304
Extended Sequence Number (ESN) Addendum to IPsec Domain of Interpretation (DOI) for Internet Security Association and Key Management Protocol (ISAKMP) S. Kent

RFC 4307
Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2) J. Schiller

RFC 4308
Cryptographic Suites for IPsec P. Hoffman

RFC 4434
The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol P. Hoffman

RFC 4443
Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification A. Conta, S. Deering

RFC 4552
Authentication/Confidentiality for OSPFv3 M. Gupta, N. Melam

RFC 4678
Server/Application State Protocol v1 A. Bivens

RFC 4753
ECP Groups for IKE and IKEv2 D. Fu, J. Solinas

RFC 4754
IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA) D. Fu, J. Solinas

RFC 4809
Requirements for an IPsec Certificate Management Profile C. Bonatti, Ed., S. Turner, Ed., G. Lebovitz, Ed.

RFC 4835
Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH) V. Manral

RFC 4862
IPv6 Stateless Address Autoconfiguration S. Thomson, T. Narten, T. Jinmei

RFC 4868
Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec S. Kelly, S. Frankel

RFC 4869
Suite B Cryptographic Suites for IPsec L. Law, J. Solinas

RFC 4941
Privacy Extensions for Stateless Address Autoconfiguration in IPv6 T. Narten, R. Draves, S. Krishnan

RFC 4945
The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX B. Korver

RFC 5014
IPv6 Socket API for Source Address Selection E. Nordmark, S. Chakrabarti, J. Laganier

RFC 5095
Deprecation of Type 0 Routing Headers in IPv6 J. Abley, P. Savola, G. Neville-Neil

RFC 5175
IPv6 Router Advertisement Flags Option B. Haberman, Ed., R. Hinden

RFC 5282
Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol D. Black, D. McGrew

RFC 5996
Internet Key Exchange Protocol Version 2 (IKEv2) C. Kaufman, P. Hoffman, Y. Nir, P. Eronen

RFC 7627
Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension K. Bhargavan, A. Delignat-Lavaud, A. Pironti, Inria Paris-Rocquencourt, A. Langley, M. Ray

RFC 8446
The Transport Layer Security (TLS) Protocol Version 1.3 E. Rescorla

Internet drafts

Internet drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Other groups can also distribute working documents as Internet drafts. You can see Internet drafts at http://www.ietf.org/ID.html.

Appendix F. Accessibility

Publications for this product are offered in Adobe Portable Document Format (PDF) and should be compliant with accessibility standards. If you experience difficulties when using PDF files, you can view the information through the z/OS Internet Library website http://www.ibm.com/systems/z/os/zos/library/bkserv/ or IBM Documentation https://www.ibm.com/docs/en. If you continue to experience problems, send a message to Contact z/OS web page(www.ibm.com/systems/z/os/zos/webqs.html) or write to:

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. See z/OS TSO/E Primer, z/OS TSO/E User's Guide, and z/OS ISPF User's Guide Vol I for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

One exception is command syntax that is published in railroad track format, which is accessible using screen readers with IBM Documentation, as described in "Dotted decimal syntax diagrams" on page 163.

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users accessing IBM Documentation using a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line, because they can be considered as a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that your screen reader is set to read out punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, you know that your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol can be used next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol giving information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, this indicates a reference that is defined elsewhere. The string following the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you should see separate syntax fragment OP1.

The following words and symbols are used next to the dotted decimal numbers:

- A question mark (?) means an optional syntax element. A dotted decimal number followed by the ? symbol indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that syntax elements NOTIFY and UPDATE are optional; that is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.
- An exclamation mark (!) means a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the same dotted decimal number can specify a ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the default option for the FILE keyword. In this example, if you include the FILE keyword but do not specify an option, default option KEEP will be applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.
- An asterisk (*) means a syntax element that can be repeated 0 or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3*, 3 HOST, and 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

  Notes:

  1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
  2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you could write HOST STATE, but you could not write HOST HOST.
  3. The * symbol is equivalent to a loop-back line in a railroad syntax diagram.

- + means a syntax element that must be included one or more times. A dotted decimal number followed by the + symbol indicates that this syntax element must be included one or more times; that is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can only repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loop-back line in a railroad syntax diagram.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-1785 United States of America

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbashi-Hakozakicho, Chuo-ku Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for the IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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 Site Counsel 2455 South Road Poughkeepsie, NY 12601-5400 USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include 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.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Bibliography

This bibliography contains descriptions of the documents in the z/OS Communications Server library.

z/OS Communications Server documentation is available online at the z/OS Internet Library web page at http://www.ibm.com/systems/z/os/zos/library/bkserv/.

z/OS Communications Server library updates

Updates to documents are also available on RETAIN and in information APARs (info APARs). Go to https://www.ibm.com/mysupport to view information APARs.

- z/OS Communications Server V2R1 New Function APAR Summary
- z/OS Communications Server V2R2 New Function APAR Summary
- z/OS Communications Server V2R3 New Function APAR Summary
- z/OS Communications Server V2R4 New Function APAR Summary

z/OS Communications Server information

z/OS Communications Server product information is grouped by task in the following tables.

Planning

Title	Number	Description
z/OS Communications Server: New Function Summary	GC27-3664	This document is intended to help you plan for new IP or SNA functions, whether you are migrating from a previous version or installing z/OS for the first time. It summarizes what is new in the release and identifies the suggested and required modifications needed to use the enhanced functions.
z/OS Communications Server: IPv6 Network and Application Design Guide	SC27-3663	This document is a high-level introduction to IPv6. It describes concepts of z/OS Communications Server's support of IPv6, coexistence with IPv4, and migration issues.

Resource definition, configuration, and tuning

Title	Number	Description
z/OS Communications Server: IP Configuration Guide	SC27-3650	This document describes the major concepts involved in understanding and configuring an IP network. Familiarity with the z/OS operating system, IP protocols, z/OS UNIX System Services, and IBM Time Sharing Option (TSO) is recommended. Use this document with the z/OS Communications Server: IP Configuration Reference.

Title	Number	Description
z/OS Communications Server: IP Configuration Reference	SC27-3651	This document presents information for people who want to administer and maintain IP. Use this document with the z/OS Communications Server: IP Configuration Guide. The information in this document includes: • TCP/IP configuration data sets • Configuration statements • Translation tables • Protocol number and port assignments
z/OS Communications Server: SNA Network Implementation Guide	SC27-3672	This document presents the major concepts involved in implementing an SNA network. Use this document with the z/OS Communications Server: SNA Resource Definition Reference.
z/OS Communications Server: SNA Resource Definition Reference	SC27-3675	This document describes each SNA definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect SNA. Use this document with the z/OS Communications Server: SNA Network Implementation Guide.
z/OS Communications Server: SNA Resource Definition Samples	SC27-3676	This document contains sample definitions to help you implement SNA functions in your networks, and includes sample major node definitions.
z/OS Communications Server: IP Network Print Facility	SC27-3658	This document is for systems programmers and network administrators who need to prepare their network to route SNA, JES2, or JES3 printer output to remote printers using TCP/IP Services.

Operation

Title	Number	Description
z/OS Communications Server: IP User's Guide and Commands	SC27-3662	This document describes how to use TCP/IP applications. It contains requests with which a user can log on to a remote host using Telnet, transfer data sets using FTP, send electronic mail, print on remote printers, and authenticate network users.
z/OS Communications Server: IP System Administrator's Commands	SC27-3661	This document describes the functions and commands helpful in configuring or monitoring your system. It contains system administrator's commands, such as TSO NETSTAT, PING, TRACERTE and their UNIX counterparts. It also includes TSO and MVS commands commonly used during the IP configuration process.
z/OS Communications Server: SNA Operation	SC27-3673	This document serves as a reference for programmers and operators requiring detailed information about specific operator commands.
z/OS Communications Server: Quick Reference	SC27-3665	This document contains essential information about SNA and IP commands.

Customization

Title	Number	Description
z/OS Communications Server: SNA Customization	SC27-3666	This document enables you to customize SNA, and includes the following information: • Communication network management (CNM) routing table • Logon-interpret routine requirements • Logon manager installation-wide exit routine for the CLU search exit • TSO/SNA installation-wide exit routines • SNA installation-wide exit routines

Writing application programs

Title	Number	Description
z/OS Communications Server: IP Sockets Application Programming Interface Guide and Reference	SC27-3660	This document describes the syntax and semantics of program source code necessary to write your own application programming interface (API) into TCP/IP. You can use this interface as the communication base for writing your own client or server application. You can also use this document to adapt your existing applications to communicate with each other using sockets over TCP/IP.
z/OS Communications Server: IP CICS Sockets Guide	SC27-3649	This document is for programmers who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using z/OS TCP/IP.
z/OS Communications Server: IP IMS Sockets Guide	SC27-3653	This document is for programmers who want application programs that use the IMS TCP/IP application development services provided by the TCP/IP Services of IBM.
z/OS Communications Server: IP Programmer's Guide and Reference	SC27-3659	This document describes the syntax and semantics of a set of high-level application functions that you can use to program your own applications in a TCP/IP environment. These functions provide support for application facilities, such as user authentication, distributed databases, distributed processing, network management, and device sharing. Familiarity with the z/OS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.
z/OS Communications Server: SNA Programming	SC27-3674	This document describes how to use SNA macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain.
z/OS Communications Server: SNA Programmer's LU 6.2 Guide	SC27-3669	This document describes how to use the SNA LU 6.2 application programming interface for host application programs. This document applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this document.)
z/OS Communications Server: SNA Programmer's LU 6.2 Reference	SC27-3670	This document provides reference material for the SNA LU 6.2 programming interface for host application programs.

Title	Number	Description
z/OS Communications Server: CSM Guide	SC27-3647	This document describes how applications use the communications storage manager.

Diagnosis

Title	Number	Description
z/OS Communications Server: IP Diagnosis Guide	GC27-3652	This document explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.
z/OS Communications Server: ACF/TAP Trace Analysis Handbook	GC27-3645	This document explains how to gather the trace data that is collected and stored in the host processor. It also explains how to use the Advanced Communications Function/Trace Analysis Program (ACF/TAP) service aid to produce reports for analyzing the trace data information.
z/OS Communications Server: SNA Diagnosis Vol 1, Techniques and Procedures and z/OS Communications Server: SNA Diagnosis Vol 2, FFST Dumps and the VIT	GC27-3667 GC27-3668	These documents help you identify an SNA problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation.
z/OS Communications Server: SNA Data Areas Volume 1 and z/OS Communications Server: SNA Data Areas Volume 2	GC31-6852 GC31-6853	These documents describe SNA data areas and can be used to read an SNA dump. They are intended for IBM programming service representatives and customer personnel who are diagnosing problems with SNA.

Messages and codes

Title	Number	Description
z/OS Communications Server: SNA Messages	SC27-3671	This document describes the ELM, IKT, IST, IUT, IVT, and USS messages. Other information in this document includes: • Command and RU types in SNA messages • Node and ID types in SNA messages • Supplemental message-related information
z/OS Communications Server: IP Messages Volume 1 (EZA)	SC27-3654	This volume contains TCP/IP messages beginning with EZA.
z/OS Communications Server: IP Messages Volume 2 (EZB, EZD)	SC27-3655	This volume contains TCP/IP messages beginning with EZB or EZD.
z/OS Communications Server: IP Messages Volume 3 (EZY)	SC27-3656	This volume contains TCP/IP messages beginning with EZY.
z/OS Communications Server: IP Messages Volume 4 (EZZ, SNM)	SC27-3657	This volume contains TCP/IP messages beginning with EZZ and SNM.
z/OS Communications Server: IP and SNA Codes	SC27-3648	This document describes codes and other information that appear in z/OS Communications Server messages.

Index

Communicating your comments to IBM

Important: If your comment regards a technical question or problem, see instead "If you have a technical problem" on page 181.

Submit your feedback by using the appropriate method for your type of comment or question:

Feedback on z/OS function

If your comment or question is about z/OS itself, submit a request through the IBM RFE Community (www.ibm.com/developerworks/rfe/).

Feedback on IBM Documentation function

If your comment or question is about the IBMDocumentation functionality, for example search capabilities or how to arrange the browser view, send a detailed email to IBM Documentation Support at ibmdocs@us.ibm.com.

Feedback on the z/OS product documentation and content

If your comment is about the information that is provided in the z/OS product documentation library, send a detailed email to mhvrcfs@us.ibm.com. We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information.

To help us better process your submission, include the following information:

- Your name, company/university/institution name, and email address
- The title and order name of the document, and the version of z/OS Communications Server
- The section title of the specific information to which your comment relates
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive authority to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

If you have a technical problem or question, do not use the feedback methods that are provided for sending documentation comments. Instead, take one or more of the following actions:

- Go to the IBM Support Portal (support.ibm.com).
- Contact your IBM service representative.
- Call IBM technical support.

Product Number: 5650-ZOS

SC27-3658-50