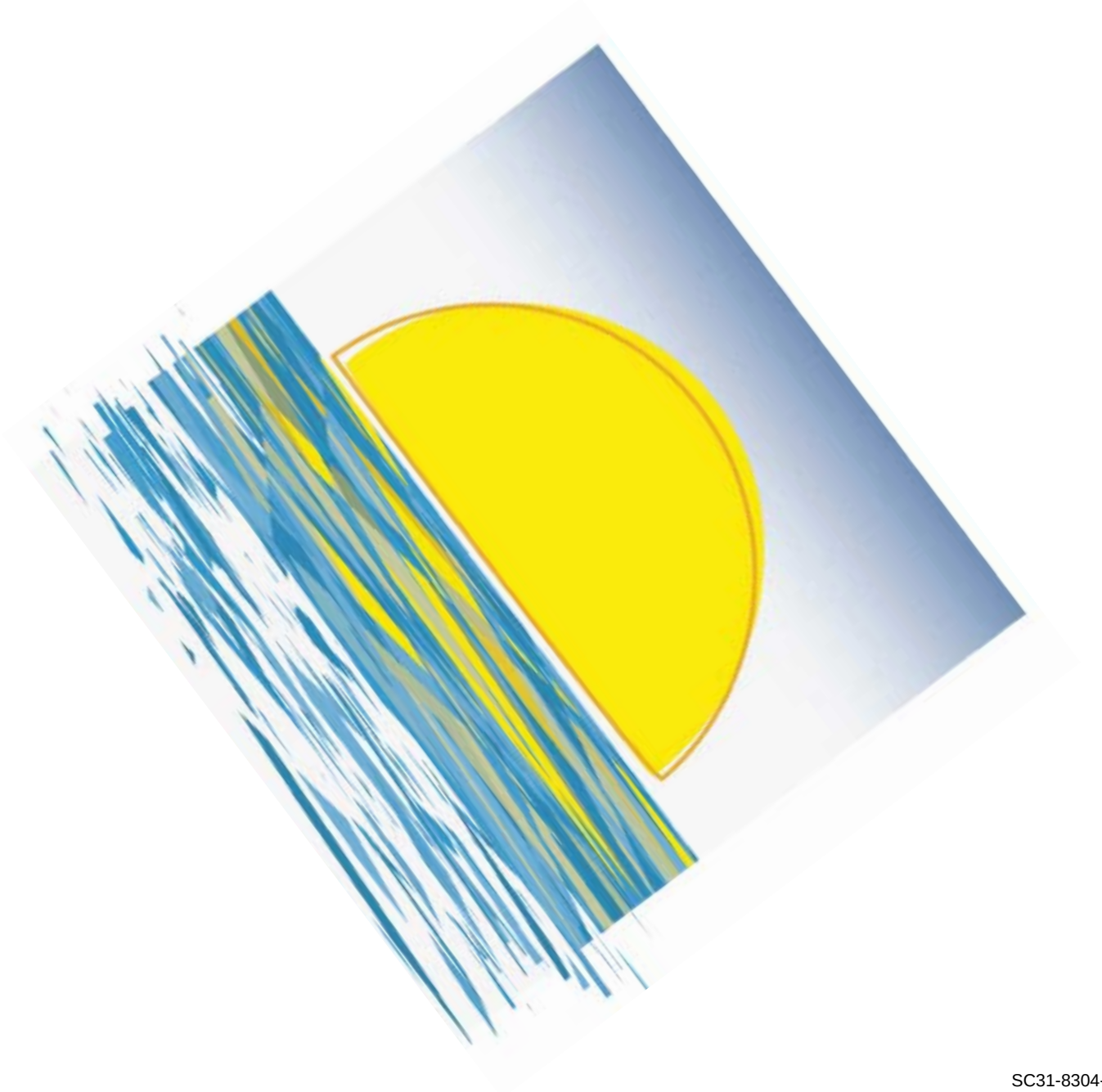
OS/390 TCP/IP OpenEdition

Configuration Guide



SC31-8304-00

OS/390 TCP/IP OpenEdition

Configuration Guide

SC31-8304-00

Note:

Before using this information and the product it supports, be sure to read the general information under Appendix F, “Notices” on page 271.

First Edition (June 1997)

This edition applies to OS/390 (5645-001) and OS/390 TCP/IP OpenEdition. See the “Summary of Changes” for a description of the changes made in this edition. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

IBM welcomes your comments. A form for readers' comments may be at the back of this publication. If the form has been removed, you may send your comments to the following address:

International Business Machines Corporation
Department CGMD
P.O. Box 12195
Research Triangle Park, North Carolina 27709
USA

If you prefer to send comments electronically, use one of the following methods:

Fax (USA and Canada):	1-800-227-5088
Internet e-mail:	usib2hpd@vnet.ibm.com
World Wide Web:	http://www.s390.ibm.com/os390
IBMLink:	CIBMORCF at RALVM13
IBM Mail Exchange:	USIB2HPD at IBMMAIL

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

Contents

Figures

Tables

About This Book

This book describes how to configure the address spaces, servers, and applications available in OS/390 TCP/IP OpenEdition. This book also describes how to customize and administer OS/390 TCP/IP OpenEdition for your specific needs.

The Network File System** (NFS**) is provided as part of the IBM MVS/Data Facility Product (MVS/DFP*).

For comments and suggestions about this book, use the comment form located at the back of this book. This form gives instructions for submitting your comments by mail, by fax, or electronically.

OS/390 TCP/IP OpenEdition is an integral part of the OS/390 family of products. For an overview and mapping of the documentation available for OS/390, see the *OS/390 Information Roadmap.*

Who Should Use This Book

This book is intended for programmers and system administrators who are familiar with TCP/IP, MVS and the IBM Time Sharing Option (TSO).

Where to Find Related Information

For information about the IBM 3172 Interconnect Controller, see:

- *IBM 3172 Interconnect Controller Installation and Service Guide*
- *IBM 3172 Interconnect Controller Operator's Guide*
- *IBM 3172 Interconnect Controller Planning Guide*
- *IBM Interconnect Controller Program User's Guide*
- *IBM 3172 Interconnect Controller Model 3 Maintenance Information*

For information about the IBM 8232 LAN Channel Station, see:

- *IBM 8232 LAN Channel Station: Operator Guide*
- *IBM 8232 LAN Channel Station: Installation and Testing*
- *IBM LAN Channel Support Program: Version 1.0 User's Guide*

For more information about the HYPERchannel** feature, see *HYPERchannel A220 Processor Adapter* 4290007, Network Systems Corporation.

You can order all the IBM books through your local IBM branch office.

Where to Find Related Information on the Internet

You may find the following information helpful.

For current updates to the TCP/IP Version 3 Release 2 for MVS documentation described in "Bibliography" on page 275, check out the TCP/IP for MVS home page:

http://www.networking.ibm.com/tcm/tcmprod.html

To keep in close touch with OS/390, we suggest you look at the OS/390 home page:

http://www.s390.ibm.com/os390

To keep abreast of new products and technologies from IBM Networking, take a look at the IBM Networking home page:

http://www.networking.ibm.com/

The IBM Networking Software Glossary is now available in HTML format as well as PDF. You can access it directly at the following URL:

http://www.networking.ibm.com/nsg/nsggls.htm

How to Contact IBM Service

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-237-5511). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

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

Summary of Changes

Summary of Changes
for SC31-8304-00

This is the first edition of this book. It contains information previously presented in *TCP/IP for MVS: Customization and Administration Guide* (SC31-7134-03), which supports TCP/IP Version 3 Release 2 for MVS. This book is new for OS/390 TCP/IP OpenEdition, which provides OpenEdition function for TCP/IP in the OS/390 environment. For information about previously available TCP/IP function, continue to use the TCP/IP Version 3 Release 2 for MVS library.

Part 1. Configuring the Base TCP/IP System

Chapter 1. Before You Begin

Before you begin to install or customize OS/390 TCP/IP OpenEdition, it is important that you:

- Review *OS/390 TCP/IP OpenEdition Planning and Release Guide*

  This book explains the differences between OS/390 TCP/IP OpenEdition and the previous release and provides the information you need to plan the installation and configuration.

- Review "Overview for New Users" in this chapter

  This table allows you to see at a glance the functions that OS/390 TCP/IP OpenEdition offers you.

- Read "Configuration Data Sets and HFS Files" on page 5 in this chapter

  This information details the way in which different functions within TCP/IP search for the data sets and HFS files they use. A thorough understanding of this process will ensure that your system has access to the correct data sets or HFS files when they are needed.

- Have a copy of *OS/390 OpenEdition Planning* handy.

  This book is referred to often.

When you have completed these steps, refer to "Customization Checklist" on page 18 to decide which chapters of this book to use. Your individual TCP/IP configuration might not require every server and function explained in this book. This checklist will guide you to the tasks and information you need.

Overview for New Users

Table 1 describes the functions TCP/IP provides and where you can find more information about them.

Table 1 (Page 1 of 3). Overview of Functions Available in OS/390 TCP/IP OpenEdition.

Function	Description	Additional Information
Connecting		
TCP/IP provides many options for connecting.	1. Channel-to-channel (CTC) devices 2. LAN channel station (LCS) devices (IBM 8232 or IBM 3172) 3. Common Link Access to Workstation (CLAW) devices (for example, RISC System/6000) 4. Virtual Devices (VIPA)	1. "DEVICE and LINK Statement—CTC Devices" on page 54 2. "DEVICE and LINK Statement—LAN Channel Station Devices" on page 56 3. " DEVICE and LINK Statement—CLAW Devices" on page 61 4. "DEVICE and LINK Statement—Virtual Devices (VIPA)" on page 64
Logging in to remote hosts		
OE Telnet	OE Telnet is a terminal emulation protocol that allows you to log on to a remote host as though you are directly attached to that host. OE Telnet allows users on any host to have access to applications running on this host. You can establish and toggle between concurrent sessions with different hosts or multiple sessions with a single host.	Chapter 6, "Configuring the OE Telnet Server" on page 125

Table 1 (Page 2 of 3). Overview of Functions Available in OS/390 TCP/IP OpenEdition.

Function	Description	Additional Information
Transferring Files		
OE File Transfer Protocol (FTP)	File transfer through OE FTP makes the entire network a resource for users to store, share, and distribute data, regardless of the mix of devices and operating systems in the network. With file transfer, you can: • Manage remote host directories, by listing, changing, creating, and deleting remote directories • Copy or move files and programs between your (local) host and remote hosts, or between two remote hosts.	Chapter 7, "Configuring the OE File Transfer Protocol (FTP) Server" on page 133
Running Programs on Remote Hosts		
OE Remote Execution (REXEC) Server and Client	With remote execution, you can send any command that is valid on a remote host and receive the results at the local host. Users can receive the results on their TSO terminals. This server runs the OE Remote EXecution Command Daemon (OREXECD), which supports both the Remote Execution (REXEC) and Remote Shell (RSH) protocols. The OREXECD server handles commands issued by remote hosts. The server performs automatic login and user authentication when user ID and password are entered.	Chapter 8, "Configuring the OE Remote Execution Server" on page 195
Managing Your Network		
OE Simple Network Management Protocol (SNMP) Server and Client	SNMP allows you to manage your TCP/IP network. SNMP provides functions for network monitoring and management. As the name implies, it is a simple protocol, minimizing the number and complexity of network management functions. SNMP is comprised of two components: a server and a client. The server is called an agent, and the client is called a manager. OS/390 TCP/IP OpenEdition supports both the SNMP manager and agent functions.	Chapter 9, "Configuring Simple Network Management Protocol (SNMP) for OE" on page 199
Querying hosts and users in the network	TCP/IP provides a number of commands for displaying information about various networks, users, and your local host. For example, you can use the oping command to determine whether your workstation can connect to a particular host and the approximate time it takes for that host to respond. You can use the onetstat command to display information about your local host, such as status of the TCP connections and devices.	See how to use the onetstat and oping commands in the *OS/390 TCP/IP OpenEdition User's Guide*
Maintaining routing tables with OE RouteD	To dynamically create and maintain network routing tables, you can use the OE RouteD function. The ORouteD server uses the RIP protocol to allow gateways and routers to create and maintain network routing tables. The ORouteD server determines whether a route is unavailable, or whether a more efficient route exists, and updates hosts' routing tables automatically.	Chapter 10, "Configuring the OE RouteD Server" on page 219

Table 1 (Page 3 of 3). Overview of Functions Available in OS/390 TCP/IP OpenEdition.

Function	Description	Additional Information
Registering the location of RPC applications with Portmapper	The Portmapper function registers the location of RPC applications. The PORTMAP address space keeps track of the relationship between RPC program numbers and the ports they are connected to.	Chapter 11, "Configuring the OE PORTMAP Address Space" on page 245

Configuration Data Sets and HFS Files

The following topics are discussed in this section:

- "Overview of Data Sets and HFS Files"
- "How OS/390 TCP/IP OpenEdition Searches for Configuration Files"
- "Information Specific to Data Sets in Configuration File Search Orders" on page 6
- "Search Order and Configuration Files for the OS/390 TCP/IP OpenEdition Stack" on page 8
- "Search Order and Configuration Files for OS/390 TCP/IP OE Applications" on page 12

Overview of Data Sets and HFS Files

Data set and *file* are comparable terms. If you are familiar with MVS, you probably use the term data set to describe a unit of data storage. If you are familiar with AIX or UNIX, you probably use the term file to describe a named set of records stored or processed as a unit. In the OS/390 TCP/IP OpenEdition environment, the files are arranged in a hierarchical file system (HFS) and are called HFS files.

Some data sets and HFS files have special importance because of their function. For example, certain data sets and HFS files are used when configuring the OS/390 TCP/IP OpenEdition environment. Other data sets, like the Telnet server (Telnet daemon) perform specific communication functions. This section describes the data sets and HFS files necessary for configuring the OS/390 TCP/IP OpenEdition environment and the search orders used to find them. A search order can include both HFS files and data sets, and these data sets and HFS files will be collectively referred to as the *configuration files* in this section.

How OS/390 TCP/IP OpenEdition Searches for Configuration Files

It is important to understand how the OS/390 TCP/IP OpenEdition environment searches for the configuration files, and when you can override the default search order with environment variables, JCL, or other variables you provide. This knowledge allows you to accommodate your local data set and HFS file naming standards, and it is helpful to know the data set or HFS file in use when diagnosing problems.

An important point to note is that what is being referred to here as the OS/390 TCP/IP OpenEdition environment consists of the OS/390 TCP/IP OpenEdition stack (OE stack) and the OS/390 TCP/IP OpenEdition applications (OE applications). Both the OE stack and OE applications have some common (or global) configuration files, but they also use configuration files that are different. The next section

provides data set specific information, and the following two sections look at the global configuration files and search orders for each of them.

Another important point to note is that when a search order is applied for any configuration file, the search ends with the first file found. Therefore, unexpected results are possible if you place configuration information in a file that never gets found due to other files existing earlier in the search order.

Information Specific to Data Sets in Configuration File Search Orders

Within the search order for the configuration files, there are data sets listed that are explicitly or dynamically (implicitly) allocated (the term allocation refers to the process of requesting access to a data set).

Explicit Data Set Allocation

Explicitly allocated data sets in a configuration file search order consist of those data sets that you specify through the use of DD statements in JCL procedures.

Dynamic Data Set Allocation

OS/390 TCP/IP OpenEdition makes extensive use of dynamically allocated data sets using the MVS Dynamic Data Set Allocation function. Multiple versions of a data set can exist, each having a different high-level qualifier or middle-level qualifier. The search order for any configuration file will determine which data set is found and used.

High-Level Qualifier: TCP/IP is distributed with a default high-level qualifier (HLQ) of *TCPIP*. This HLQ is a hard-coded character string within OS/390 TCP/IP OpenEdition.

For dynamic data set allocation, you can accept the default HLQ distributed with OS/390 TCP/IP OpenEdition or override it. To override the default HLQ used by dynamic data set allocation, specify the DATASETPREFIX statement in the PROFILE.TCPIP and TCPIP.DATA configuration files (both these files are described later). The DATASETPREFIX value is used as the last step in the search order for most configuration files. However, because DATASETPREFIX is included in PROFILE.TCPIP and TCPIP.DATA, it is not used as the last step in the search order for those configuration files.

Note: The OE stack uses the DATASETPREFIX statement found in PROFILE.TCPIP, while the OE applications use the DATASETPREFIX found in TCPIP.DATA.

Middle-Level Qualifiers: Multiple middle level qualifiers (MLQ) permit the isolation of certain profile and translation table data sets. Two of the possible middle-level qualifiers are:

- Node name

  Node name is a MLQ used in the search order for finding the configuration file PROFILE.TCPIP. Node name is determined by the parameters specified during VMCF initialization. For further information on initializing VMCF, see the *OS/390 TCP/IP OpenEdition Program Directory*.

- Function name

  The OS/390 TCP/IP OpenEdition implementation of national language support (NLS) and double-byte character set (DBCS) support requires the use of mul-

tiple translation tables. To facilitate the concurrent use of multiple languages and code pages, TCP/IP uses a middle-level qualifier to designate which server or client uses a particular translation table. STANDARD, the default MLQ, is available for use if a single translation table can be used by multiple servers or clients. The TCP/IP commands TELNET and FTP provide a TRANSLATE parameter that permits you to specify your chosen MLQ to replace the function name for that invocation of the command. For example, SRVRFTP is used as a MLQ by the OE File Transfer Protocol server.

Dynamically Allocated Data Sets: A dynamically allocated data set can have from 1 to 6 different fully-qualified data set names, depending on the function of the data set. The general naming convention and its specific application are explained in "Naming Conventions for Dynamically Allocated Data Sets."

Following are some of the data sets that can be dynamically allocated by TCP/IP in a configuration file search order (you cannot specify them with DD statements in JCL):

```
ETC.PROTO               ETC.RPC
HOSTS.ADDRINFO          HOSTS.SITEINFO
SRVRFTP.TCPCHBIN        SRVRFTP.TCPHGBIN
SRVRFTP.TCPKJBIN        SRVRFTP.TCPSCBIN
SRVRFTP.TCPXLBIN        STANDARD.TCPCHBIN
STANDARD.TCPHGBIN       STANDARD.TCPKJBIN
STANDARD.TCPSCBIN       STANDARD.TCPXLBIN
```

For each of these data sets, the fully-qualified name is established by using one of the following values as the data set HLQ:

- User ID or job name
- DATASETPREFIX value

Naming Conventions for Dynamically Allocated Data Sets: A data set that you create for the purpose of being allocated dynamically by TCP/IP must use the following naming conventions. A data set that you allocate explicitly (with a DD statement in JCL) can have any valid MVS data set name or HFS file name.

- *userid.yyyy.zzzz*

  userid is the user ID of the logged on TSO user.

- *TSOprefix.yyyy.zzzz*

  TSOprefix is the data set prefix established by the TSO PROFILE command. *userid* is the default value of *TSOprefix*

- *jobname.yyyy.zzzz*

  jobname is the job name specified on the JOB statement for a job stream or the procedure name for a started procedure.

- *hlq.yyyy.zzzz*

  hlq is the TCP/IP HLQ distributed as the system default, which can be overridden by the value in the DATASETPREFIX statement.

- *xxxx.nodename.zzzz*

  nodename is an MLQ that is used to define the data set name for the TCP/IP OE stack profile data set.

- *xxxx.function_name.zzzz*

  function_name denotes an acronym specifying a particular TCP/IP server (for example SRVRFTP for the OE FTP server) and is used as a MLQ for the translation table data set for that application.

- *xxxx.private_name.zzzz*

  private_name is a user-specified private qualifier that can be specified as an option on some TCP/IP commands.

- SYS1.TCPPARMS(TCPDATA)

  member of a system data set used to find the *configuration file* TCPIP.DATA. (You can allocate the SYS1.TCPPARMS data set with partitioned organization (PO), a fixed block format (FB), a logical record length of 80, and any valid blocksize for a fixed block, such as 3120.

Search Order and Configuration Files for the OS/390 TCP/IP OpenEdition Stack

Two configuration files are used by the OE stack. They are PROFILE.TCPIP and TCPIP.DATA. PROFILE.TCPIP is used only for the configuration of the OE stack. TCPIP.DATA is used during configuration of both the OE stack and the OE applications; the search order used to find TCPIP.DATA is the same for both the OE stack and applications.

PROFILE.TCPIP Search Order

During initialization of the OE stack (also referred to in this book as the TCPIP address space or the OE TCPIP address space), system operation and configuration parameters for the OE stack are read from the configuration file PROFILE.TCPIP. The search order used by the OE stack to find PROFILE.TCPIP involves both explicit and dynamic data set allocation as follows:

- //PROFILE DD DSN=*aaa.bbb.ccc(anyname)*

  Explicitly specifying the PROFILE DD statement in the TCPOPROC JCL is the recommended way to specify PROFILE.TCPIP. If this DD statement is present, the data set it defines is explicitly allocated by MVS. No dynamic allocation is done. The rest of the data sets looked for in the PROFILE.TCPIP search order are dynamically allocated by TCP/IP.

- *jobname.nodename*.TCPIP
- *hlq.nodename*.TCPIP
- *jobname*.PROFILE.TCPIP
- *hlq*.PROFILE.TCPIP

Remember that for the above PROFILE.TCPIP search order, the default *hlq* distributed with TCPIP is the string *TCPIP*. For detailed information about the content and specification of the configuration file PROFILE.TCPIP, see Chapter 3, “Configuring the TCPIP Address Space” on page 33.

TCPIP.DATA Search Order

During initialization of the OE stack, the configuration file TCPIP.DATA is also used. The search order used to find TCPIP.DATA is as follows:

- The MVS data set or HFS file that is identified in an environment variable called RESOLVER_CONFIG.

  This environment variable is passed as a parameter to the OE stack in the TCPOPROC JCL used to start TCPIP. The following is an example of specifying this environment variable in the JCL:

```
//TCPIP33X  PROC
// PARMS='ENVAR("RESOLVER_CONFIG=/etc/tcpv33a.data") CTRACE(CTIEZB00)'
//*
//* OS/390 TCP/IP OpenEdition
//* SMP/E Distribution Name: EZBOPROC
//*
//*     5645-001 5655-HAL (C) Copyright IBM Corp. 1989, 1997.
//*     All rights reserved.
//*     US Government Users Restricted Rights -
//*     Use, duplication or disclosure restricted
//*     by GSA ADP Schedule Contract with IBM Corp.
//*     See IBM Copyright Instructions
//*
//TCPIP    EXEC PGM=EZBTCPIP,
//             PARM='&PARMS',
//             REGION=7500K,TIME=1440
//*
```

- /etc/resolv.conf

  This is the file /etc/resolv.conf that resides in the HFS.

- //SYSTCPD DD DSN=*ddd.eee.fff(anyname)*

  The //SYSTCPD DD card can be specified in TCPOPROC JCL.

- *jobname*.TCPIP.DATA

- SYS1.TCPPARMS(TCPDATA)

  For more information on this data set, see “Naming Conventions for Dynamically Allocated Data Sets” on page 7.

- *hlq*.TCPIP.DATA

  Remember that the default *hlq* distributed with TCPIP is the string *TCPIP*. So, effectively, this is TCPIP.TCPIP.DATA in the search order.

For a single OpenEdition TCP/IP instance, the HFS file /etc/resolv.conf can be safely used. In a multiple OpenEdition TCP/IP instance environment, the environment variable RESOLVER_CONFIG should be used. If the SYSTCPD DD is used, remember that it is only searched for if the HFS file /etc/resolv.conf does not exist. For more information about a multiple OpenEdition TCP/IP instance environment, see “Considerations for Multiple Instances of TCP/IP” on page 26.

Examples

The following examples show the search order used by TCP/IP to find the configuration files PROFILE.TCPIP and TCPIP.DATA.

Search Order When DD Cards Are In Your TCPIP Startup Procedure: In this example, the PROFILE and SYSTCPD DD cards are specified in the TCPOPROC JCL for TCP/IP as follows:

```
//TCPIP33X  PROC PARMS='CTRACE(CTIEZB00)'
//*
//* OS/390 TCP/IP OpenEdition
//* SMP/E Distribution Name: EZBOPROC
//*
//*     5645-001 5655-HAL (C) Copyright IBM Corp. 1989, 1997.
//*     All rights reserved.
//*     US Government Users Restricted Rights -
//*     Use, duplication or disclosure restricted
//*     by GSA ADP Schedule Contract with IBM Corp.
//*     See IBM Copyright Instructions
//*
//TCPIP    EXEC PGM=EZBTCPIP,
//             PARM='&PARMS',
//             REGION=7500K,TIME=1440
//*
⋮
//PROFILE DD DISP=SHR,DSN=MVSA.PROD.PARMS(PROFILE)
⋮
//SYSTCPD DD DISP=SHR,DSN=MVSA.PROD.PARMS(TCPDATA)
⋮
```

For the configuration file PROFILE.TCPIP, since the PROFILE DD is the first step in the search sequence, the data set MVSA.PROD.PARMS(PROFILE) is used. For the configuration file TCPIP.DATA, since the environment variable RESOLVER_CONFIG is not being passed as a parameter to TCP/IP in TCPOPROC JCL, whether SYSTCPD DD is used or not depends on if the HFS file /etc/resolv.conf exists. If it exists, TCP/IP will use that file; otherwise, the data set MVSA.PROD.PARMS(TCPDATA) is used.

Search Order When No DD Cards Are In Your TCP/IP Startup Procedure: For this example, the TCPOPROC JCL has neither the PROFILE nor SYSTCPD DD cards:

```
//TCPIP33X  PROC PARMS='CTRACE(CTIEZB00)'
//*
//* OS/390 TCP/IP OpenEdition
//* SMP/E Distribution Name: EZBOPROC
//*
//*     5645-001 5655-HAL (C) Copyright IBM Corp. 1989, 1997.
//*     All rights reserved.
//*     US Government Users Restricted Rights -
//*     Use, duplication or disclosure restricted
//*     by GSA ADP Schedule Contract with IBM Corp.
//*     See IBM Copyright Instructions
//*
//TCPIP    EXEC PGM=EZBTCPIP,
//             PARM='&PARMS',
//             REGION=7500K,TIME=1440
//*
⋮
```

For the configuration file PROFILE.TCPIP, the search sequence used is as follows:

1. PROFILE DD

   No PROFILE DD exists.

2. *jobname.nodename.*TCPIP

   If TCPIP33X.*nodename.*TCPIP is found, the search is stopped here.

3. *hlq.nodename.*TCPIP

   If TCPIP.*nodename.*TCPIP is found, the search is stopped here.

4. *jobname.*PROFILE.TCPIP

   If TCPIP33X.PROFILE.TCPIP is found, the search is stopped here.

5. *hlq.*PROFILE.TCPIP

   TCPIP.PROFILE.TCPIP is searched last if necessary.

For the configuration file TCPIP.DATA, the search sequence used is as follows:

1. Value of environment variable RESOLVER_CONFIG

   No RESOLVER_CONFIG environment variable is set.

2. /etc/resolv.conf

   If this HFS file is found, the search is stopped here

3. SYSTCPD DD

   No SYSTCPD DD exists.

4. SYS1.TCPPARMS(TCPDATA)

   If this data set is found, the search is stopped here.

5. *hlq.*TCPIP.DATA

   TCPIP.TCPIP.DATA is searched last if necessary.

Search Order and Configuration Files for OS/390 TCP/IP OE Applications

This section describes the configuration files used in common (that is, globally) by the OE applications, and the search orders for those configuration files. Each OE application can have its own configuration files that are specific for that application. For information regarding those configuration files, see the descriptions of the individual OE applications in Part 2, “Configuring the Servers” on page 123.

The configuration files commonly referenced by the OE applications are:

- TCPIP.DATA
- STANDARD.TCPXLBIN
- HOSTS.SITEINFO
- HOSTS.ADDRINFO
- ETC.PROTO
- ETC.SERVICES

Remember that, for the search orders of each of the above configuration files, a search stops when a file is found in the search order, regardless of whether what you are looking for actually exists in that file. This can be a confusing point if there is a file down the list in the search order with the actual desired value, but a previous file in the search order exists. In this case, what you expect to find is not actually found since the search ended before it got to the correct file.

Also, some of the search orders described below include values of environment variables. How to set an environment variable so that an OE application is able to retrieve the value depends on whether the OE application is started from the OE shell or from JCL.

If the OE application is to be started from the OE shell, the *export* shell command can be used to set the environment variable. For example, to set the value of RESOLVER_CONFIG to the HFS file /etc/tcpv33a.data, you can code the following export command:

```
export RESOLVER_CONFIG=/etc/tcpv33a.data
```

If instead of an HFS file, you want to set RESOLVER_CONFIG to the data set MVSA.PROD.PARMS(TCPDATA) you can specify the following export command (be sure to put the single quotes around the data set name - if you don't your user ID will be added as a prefix to the data set name when TCP/IP tries to open the file):

```
export RESOLVER_CONFIG="//'MVSA.PROD.PARMS(TCPDATA)'"
```

If the OE application is to be started from JCL instead of from the OE shell, the environment variable needs to be passed as a parameter in that OE application's JCL. For example:

```
//FTPD   PROC MODULE='FTPD',PARMS=''
//*********************************************************************
//*                                                                   *
//*         TCP/IP for MVS                                            *
//*                                                                   *
//*         Descriptive Name:            FTP Server Start Procedure   *
//*                                                                   *
//*         File Name:                   tcpip.SEZAINST(EZAFTPAP)     *
//*                                      tcpip.SEZAINST(FTPD)         *
//*                                                                   *
//*         SMP/E Distribution Name:     EZAFTPAP                     *
//*                                                                   *
//*                                                                   *
//*             Licensed Materials - Property of IBM                  *
//*             This product contains "Restricted Materials of IBM"   *
//*             5645-001 5655-HAL (C) Copyright IBM Corp. 1995, 1997. *
//*             All rights reserved.                                  *
//*             US Government Users Restricted Rights -               *
//*             Use, duplication or disclosure restricted by          *
//*             GSA ADP Schedule Contract with IBM Corp.              *
//*             See IBM Copyright Instructions.                       *
//*                                                                   *
//*                                                                   *
//*********************************************************************
//FTPD   EXEC PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//      PARM='POSIX(ON) ALL31(ON)/&PARMS'
⋮
```

TCPIP.DATA

This file is referenced to determine, among other things, the data set prefix (DATASETPREFIX keyword) to be used when trying to access the rest of the configuration files specified in this section.

The following is the search order used to find the TCPIP.DATA configuration file:

1. The MVS data set or HFS file that is identified as the value of the environment variable RESOLVER_CONFIG

   The value of the environment variable is used to fopen() the configuration file. All OpenEdition I/O rules apply. For a complete discussion, see the chapter on "Performing HFS I/O Operations" in the *OS/390 C/C++ Programming Guide*.

2. /etc/resolv.conf that resolves in the HFS
3. Any data set that is explicitly allocated to DD-name SYSTCPD (for example, using the //SYSTCPD DD card in JCL)
4. *jobname*.TCPIP.DATA for batch jobs, or *userid*.TCPIP.DATA for TSO users or OE shell users
5. SYS1.TCPPARMS(TCPDATA)
6. *hlq*.TCPIP.DATA (The default *hlq* distributed with TCP/IP is *TCPIP*).

If you intend to run multiple instances of OpenEdition TCPIP, you might want to carefully read “Considerations for Multiple Instances of TCP/IP” on page 26. That section describes how to set up your search order for TCPIP.DATA in that environment.

STANDARD.TCPXLBIN

This file is referenced to determine the translate data sets to be used.

The search order used to access this configuration file is:

1. The value of the environment variable X_XLATE

   The value of the environment variable is used to fopen() the configuration file. All OpenEdition I/O rules apply. For a complete discussion, see the chapter on "Performing HFS I/O Operations" in the *OS/390 C/C++ Programming Guide*.

2. *hlq*.STANDARD.TCPXLBIN

   hlq represents the value of the DATASETPREFIX keyword specified in the TCPIP.DATA configuration file (if found); otherwise, *hlq* is TCPIP by default.

HOSTS.SITEINFO

This file supplies the information for the following four network host database functions: gethostbyname(), sethostent(), gethostent(), and endhostent. Additionally, it supplies information for the network database function getnetbyname().

The search order used to access this configuration file is:

1. The value of the environment variable X_SITE

   The value of the environment variable is used as is to fopen() the configuration file. All OpenEdition I/O rules apply. For a complete discussion, see the chapter on "Performing HFS I/O Operations" in the *OS/390 C/C++ Programming Guide*.

   The only valid data set identified by this environment variable must contain the HOSTS.SITEINFO information created by the MAKESITE command.

   It is not recommended that an X_SITE refer to an HFS file, because the two types of data sets are incompatible.

2. /etc/hosts that resides in the HFS

3. *userid*.HOSTS.SITEINFO for TSO/E or *jobname*.HOSTS.SITEINFO for batch requests

4. *hlq*.HOSTS.SITEINFO

   hlq represents the value of the DATASETPREFIX keyword specified in the TCPIP.DATA configuration file (if found); Otherwise, *hlq* is TCPIP by default.

HOSTS.ADDRINFO

This file supplies the information for the following four network database functions: getnetbyaddr(), setnetent(), getnetent(), and endnetent(). Additionally, it supplies information for the network host database function gethostbyaddr().

The search order used to access this configuration file is:

1. The value of the environment variable X_ADDR

   The value of the environment variable is used as is to fopen() the configuration file. All OpenEdition I/O rules apply. For a complete discussion, see the chapter on "Performing HFS I/O Operations" in the *OS/390 C/C++ Programming Guide*.

   The only valid data set identified by this environment variable must contain the HOSTS.ADDRINFO information created by the MAKESITE command.

2. /etc/hosts, only if the request was gethostbyaddr(); otherwise, this step is skipped. /etc/hosts resides in the HFS.
3. *userid*.HOSTS.ADDRINFO for TSO/E and *jobname*.HOSTS.ADDRINFO for batch requests
4. *hlq*.HOSTS.ADDRINFO

   hlq represents the value of the DATASETPREFIX keyword specified in the TCPIP.DATA configuration file (if found); Otherwise, *hlq* is TCPIP by default.

ETC.PROTO

This file supplies the information for the following five protocol database functions: getprotobynumber(), getprotobyname(), setprotoent(), getprotoent(), and endprotoent().

The search order used to access this configuration file is:

1. /etc/protocol that resides in the HFS
2. *userid*.ETC.PROTO for TSO/E or *jobname*.ETC.PROTO for batch requests
3. *hlq.ETC.PROTO*

   hlq represents the value of the DATASETPREFIX keyword specified in the TCPIP.DATA configuration file (if found); Otherwise, *hlq* is TCPIP by default.

ETC.SERVICES

This file supplies the information for these five services database functions: getservbyport(), getservbyname(), setservent(), getservent(), and endservent().

The search order used to access this configuration file is:

1. /etc/services that resides in the HFS
2. *userid*.ETC.SERVICES for TSO/E or *jobname*.ETC.SERVICES for batch requests
3. hlq.ETC.SERVICES

   hlq represents the value of the DATASETPREFIX keyword specified in the TCPIP.DATA configuration file (if found); Otherwise, *hlq* is TCPIP by default.

Examples

The following example shows the OE FTP server start procedure, which indirectly requests information from common (that is, global) configuration files TCPIP.DATA, HOSTS.SITEINFO, HOSTS.ADDRINFO, and ETC.SERVICES. The OE FTP server does not search for these files directly, but calls services that search for them. For this example, the EZAFTPAP JCL that is used is as follows:

```
//FTPD   PROC MODULE='FTPD',PARMS=''
//*********************************************************************
//*                                                                   *
//*         TCP/IP for MVS                                            *
//*                                                                   *
//*         Descriptive Name:            FTP Server Start Procedure   *
//*                                                                   *
//*         File Name:                   tcpip.SEZAINST(EZAFTPAP)     *
//*                                      tcpip.SEZAINST(FTPD)         *
//*                                                                   *
//*         SMP/E Distribution Name:     EZAFTPAP                     *
//*                                                                   *
//*                                                                   *
//*             Licensed Materials - Property of IBM                  *
//*             This product contains "Restricted Materials of IBM"   *
//*             5645-001 5655-HAL (C) Copyright IBM Corp. 1995, 1997. *
//*             All rights reserved.                                  *
//*             US Government Users Restricted Rights -               *
//*             Use, duplication or disclosure restricted by          *
//*             GSA ADP Schedule Contract with IBM Corp.              *
//*             See IBM Copyright Instructions.                       *
//*                                                                   *
//*                                                                   *
//*********************************************************************
//FTPD   EXEC PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//      PARM='POSIX(ON) ALL31(ON)/&PARMS'
⋮
//*      SYSTCPD explicitly identifies which file is to be
//*      used to obtain the parameters defined by TCPIP.DATA.
//*      The SYSTCPD DD statement should be placed in the JCL of
//*      the server.  The file can be any sequential data set,
//*      member of a partitioned data set (PDS), or HFS file.
//*SYSTCPD DD DISP=SHR,DSN=TCPIP.SEZAINST(TCPDATA)
⋮
```

So, the search sequence for each configuration file is as follows:

- TCPIP.DATA
  1. Environment variable RESOLVER_CONFIG

     Since the OE FTP server is being started from JCL (rather than from the OE shell), this environment variable would have to be passed as a parameter to FTPD. Since RESOLVER_CONFIG is not passed by this JCL, the search goes to the next step in the sequence.

  2. /etc/resolv.conf

     If this HFS file exists, the search stops here.

  3. The data set specified on the SYSTCPD DD card

     Since the SYSTCPD DD card is commented out in the above JCL, the search goes to the next step in the sequence.

  4. *jobname.*TCPIP.DATA

     If FTPD.TCPIP.DATA is found, the search stops here.

  5. SYS1.TCPPARMS(TCPDATA)

     If this data set is found, the search stops here.

6. *hlq*.TCPIP.DATA

   TCPIP.TCPIP.DATA is searched last, if necessary.

- HOSTS.SITEINFO

  1. The value of the environment variable X_SITE

     Since the X_SITE environment variable was not passed in the JCL, the search goes to the next step in the sequence.

  2. /etc/hosts that resides in the HFS

  3. *jobname.*HOSTS.SITEINFO

     If FTPD.HOSTS.SITEINFO is found, the search stops here.

  4. *hlq*.HOSTS.SITEINFO

     The *hlq* used depends on the DATASETPREFIX statement found in the configuration file TCPIP.DATA. If no DATASETPREFIX statement is specified in TCPIP.DATA, the default value TCPIP is used and TCPIP.HOSTS.SITEINFO is searched for last.

- HOSTS.ADDRINFO

  1. The value of the environment variable X_ADDR

     Since the X_ADDR environment variable was not passed in the JCL, the search goes to the next step in the sequence.

  2. /etc/hosts that resides in the HFS

     If this HFS file exists, the search stops here.

  3. *jobname.*HOSTS.ADDRINFO

     If FTPD.HOSTS.ADDRINFO is found, the search stops here.

  4. *hlq*.HOSTS.ADDRINFO

     The *hlq* used depends on the DATASETPREFIX statement found in the configuration file TCPIP.DATA. If no DATASETPREFIX statement is specified in TCPIP.DATA, the default value TCPIP is used and TCPIP.HOSTS.ADDRINFO is searched for last.

- ETC.SERVICES

  1. /etc/services that resides in the HFS

     If this HFS file exists, the search stops here.

  2. *jobname.*ETC.SERVICES

     If FTPD.ETC.SERVICES is found, the search stops here.

  3. *hlq*.ETC.SERVICES

     The *hlq* used depends on the DATASETPREFIX statement found in the configuration file TCPIP.DATA. If no DATASETPREFIX statement is specified in TCPIP.DATA, the default value TCPIP is used and TCPIP.ETC.SERVICES is searched for last.

Customization Checklist

Your individual TCP/IP configuration might not have every server and function explained in this book. Use the following checklist to decide which tasks you have to do to customize your system and which chapters of this book to use. It shows when each task is required, where in the book it is explained, and which sample data sets and procedures it uses. You can find all the samples in *hlq*.SEZAINST.

Table 2 (Page 1 of 2). Checklist of Customization Tasks

Req	When	Task and Reference	Sample Data Sets	Sample Procedures
√	Always	Planning Customization of OS/390 TCP/IP OpenEdition Chapter 2 on page 21	VTAMLST	
√	Always	Configuring the TCPIP Address Space Chapter 3 on page 33	SAMOPROF	TCPOPROC
√	Always	Defining the TCP/IP Client System Parameters Chapter 4 on page 99	TCPDATA	
√	Always	Configuring the Site Table Chapter 5 on page 115	HOSTS	
	If you want to use Telnet to allow users on any host to have access to applications running on this host	Configuring the OE Telnet Server Chapter 6 on page 125	VTAMLST SAMOPROF	TCPOPROC
	If you will be using the OE File Transfer Program (FTP) Server to send or receive files across the network	Configuring the OE File Transfer Program (FTP) Server Chapter 7 on page 133	FTCDATA FTPSDATA	EZAFTPAP
	If you will be using the OE Remote Execution Server to execute TSO commands that have been received from a remote host	Configuring the OE Remote Execution Server Chapter 8 on page 195		RXPROC
	If you will be using OE Simple Network Management Protocol (SNMP) to manage TCP/IP agents in the network	Configuring Simple Network Management Protocol for OE Chapter 9 on page 199	/etc/snmpv2.conf /etc/mibs.data /etc/pw.src.cli PW.SRC SNMPTRAP.DEST OSNMPD.DATA	OSNMPDPR
	If you will be using the OE RouteD server to use the RIP protocol to allow gateways and routers to create and maintain network routing tables	Configuring the OE RouteD Server Chapter 10 on page 219	SERVICES	ROUTED

Table 2 (Page 2 of 2). Checklist of Customization Tasks

Req	When	Task and Reference	Sample Data Sets	Sample Procedures
	If you will be using the OE Portmapper function to register the location of RPC applications	Configuring the OE PORTMAP Address Space Chapter 11 on page 245		OPORTPRC

Chapter 2. Customization and Administration Overview

Before You Configure...:

Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

The following section provides a high-level overview of the procedures for customizing and administering OS/390 TCP/IP OpenEdition. It includes information on:

- Updating your procedure library (PROCLIB)
- Starting and stopping the servers and the TCP/IP address space
- Controlling the servers interactively with VARY TCPIP
- Running multiple instances (or copies) of TCP/IP

General Customization Procedure

Steps to Install and Customize OS/390 TCP/IP OpenEdition:

1. Install OS/390 TCP/IP OpenEdition
2. Customize OS/390 TCP/IP OpenEdition
3. Test and verify your configuration
4. Accept the product configuration

If you plan to use OpenEdition sockets, see *OS/390 OpenEdition Planning* for more information about setting up your system. For information about the differences between the non-OS/390 TCP/IP OpenEdition applications and the OS/390 TCP/IP OpenEdition applications, see the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

Step 1: Install OS/390 TCP/IP OpenEdition

Before you begin the installation:

- Be sure you understand the data set naming conventions used in TCP/IP. You can find this information in “Configuration Data Sets and HFS Files” on page 5.
- Consult the Program Directory for current information about the material, procedures, and storage estimates for this version.

The instructions for installing OS/390 TCP/IP OpenEdition are provided in the Program Directory shipped with the product. Follow the steps detailed in the Program Directory to prepare your system and install TCP/IP. When appropriate, the Program Directory will direct you to this book to customize the TCP/IP data sets and procedures and verify their configuration. When these tasks are complete, you can return to the Program Directory to accept the TCP/IP installation.

Step 2: Customize OS/390 TCP/IP OpenEdition

To customize TCP/IP you need to update the cataloged procedures and configuration data sets for the TCPIP address space, its clients, and servers.

OS/390 TCP/IP OpenEdition runs as a started task in its own address space. Each of the servers runs in its own address space and is started with its own procedure. The TCPIP address space requires:

- A cataloged procedure in a system or recognized PROCLIB.
- A data set that provides configuration definitions for the TCPIP address space and includes statements affecting many of the servers. This data set is referred to as PROFILE.TCPIP.
- A data set to provide the parameters that are common across all clients. This data set is referred to as TCPIP.DATA.

Many of the servers also require other data sets for their specific functions.

The subsequent chapters in this book show you how to:

- Configure the TCPIP address space by updating the samples provided in *hlq*.SEZAINST(SAMOPROF) and *hlq*.SEZAINST(TCPOPROC)
- Configure the universal client parameters provided in *hlq*.SEZAINST(TCPDATA)
- Configure the site table, defined in *hlq*.HOSTS.LOCAL, to identify the internet names and addresses of your TCP/IP host
- Customize the TCP/IP Component Trace parameters by updating the CTRACE parameter in the PARM= field of the EXEC JCL statement in the TCP/IP started procedure. See “Specifying the CTRACE Keyword” on page 35 for instructions.

  (You can find a description of the MVS Component Trace support in the *OS/390 TCP/IP OpenEdition Diagnosis Guide*.)
- Specify the ENVAR parameter on the PARMS=CTRACE(CTIEZB00) keyword to override the resolver file. For more information on setting the environment variable RESOLVER_CONFIG using the ENVAR parameter, see “Considerations for Multiple Instances of TCP/IP” on page 26.
- Configure each of the servers you want to run. This might require:
  - Modifying sample procedures and adding them in your PROCLIB
  - Modifying the configuration data set, PROFILE.TCPIP
  - Adding port numbers to *hlq*.ETC.SERVICES
  - Modifying other data sets containing server-specific parameters

Use the checklist provided in Table 2 on page 18 to decide which servers to configure and which samples they require. You can find the sample procedures and data sets in *hlq*.SEZAINST or the HFS. Table 3 on page 24 and Table 4 on page 26 provide additional reference information you can use as you configure and customize each server.

You can find general information about starting, stopping, and dynamically controlling the servers in “Administration Overview” on page 30. Specific information about operating and administering each server is also provided in the chapter for that server.

Step 3: Test and Verify Your Configuration

To verify that your configuration is correct, start the TCPIP address space.

Start the TCPIP Address Space

Enter the MVS START command from the operator's console to start TCP/IP, specifying the member name of your cataloged procedure. For example, if the procedure to start the TCPIP address space was called TCPV3R3 in your PROCLIB, you would enter:

```
START TCPV3R3
```

Step 4: Accept the Product Installation

Instructions for accepting the product installation are in the Program Directory. After you have verified your TCP/IP configuration, return to the Program Directory and follow the instructions provided there.

Cataloged Procedures and Configuration Data Sets

Table 3 on page 24 lists the cataloged procedures used by the TCP/IP functions and shows how each procedure gets the parameters it needs. These parameters can be passed directly in the procedure or they can come from configuration data sets and other data sets in the system.

The names of some of these data sets can be explicitly allocated using the JCL statements shown. The names of other data sets are hard-coded in the TCP/IP programs and, except for the high-level qualifier, cannot be changed. Table 4 on page 26 provides additional information about the configuration data sets.

Updating Your Procedure Library

Depending on your TCP/IP configuration and MVS system, you might not need to install all of the procedures shown in Table 3 on page 24. Additional information about each procedure is provided in the chapter for that server or function.

As you configure the TCPIP address space and each server, examine the sample procedures, copy them into your system PROCLIB or a recognized PROCLIB, and modify them to suit your installation.

Naming Your Procedure Members

Be aware that the name on the PORT statement in *hlq*.PROFILE.TCPIP must match the member name of the cataloged procedure you use to start that server or address space.

If you copy the sample procedure to a member with the same name as the one appearing on the PROC statement, the name on the PORT statement in SAMOPROF will not need modification.

Specifying Job Step Wait Time

The sample JCL procedures use TIME=1440 as a parameter on the EXEC statement to override Job Step Wait Timing. This is done to avoid unwanted system 522 abends.

Individual TCP/IP server and client programs may enter extended idle periods depending on the amount and type of user activity. For example, if TIME=10 was specified on the EXEC statement of the FTP procedure and no client requests occur for FTP service for more than ten minutes, the FTP server would be ended with a system 522 abend. Again, if TIME=2 were specified on the EXEC statement of the TSO procedure used for TCP/IP client use and a REXEC command is entered that takes longer than two minutes to execute at the remote host, the client TSO user would be ended with a system 522 abend.

You should evaluate your need for Job Step Wait Timing function and adjust or accept the TIME=1440 parameter, as necessary, for your installation.

Specifying Region Size

Some of the sample procedures specify a region size of 7500KB. Decreasing these values can affect the capacity of the MVS system.

Cataloged Procedures

Table 3 (Page 1 of 2). TCP/IP Cataloged Procedures

Function	Cataloged Procedures		
OE FTP	Procedure name:	EZAFTPAP	
	Sample:	SEZAINST(EZAFTPAP)	
	Purpose:	Starts the OE FTP server.	
	Parameters:	Passed in the following configuration data set:	
		hlq.FTP.DATA	
			Sample: SEZAINST(FTCDATA) - client SEZAINST(FTSDATA) - server
			Optional, can be dynamically or explicitly allocated. Recommend you explicitly allocate with //SYSFTPD DD.
OE Portmapper	Procedure name:	PORTMAP	
	Sample:	SEZAINST(OPORTPRC)	
	Purpose:	Starts the OE Portmapper server that makes RPC programs available across the network through TCP/IP.	
	Parameters:	Passed in the following configuration data sets:	
		hlq.ETC.RPC	Sample: SEZAINST(ETCRPC). Required, cannot be explicitly allocated.
OE Remote Execution Server	Procedure name:	RXSERVE	
	Sample:	SEZAINST(RXPROC)	
	Purpose:	Starts the server that executes the OE REmote EXecution Command Daemon.	
	Parameters:	Does not use any configuration data sets. All parameters are passed on the EXEC statement of RXPROC.	

Table 3 (Page 2 of 2). TCP/IP Cataloged Procedures

Function	Cataloged Procedures		
OE RouteD	Procedure name:	OROUTED	
	Sample:	SEZAINST(OROUTED)	
	Purpose:	Starts the OE RouteD server to handle dynamic routing with RIP gateways.	
	Parameters:	Passed in the following configuration data sets:	
		ORouteD gateways	
			No sample provided. Optional. Searched for in the first file found below: - GATEWAYS_FILE environment variable - /etc/gateways - *hlq*.ETC.GATEWAYS
		hlq.ETC.SERVICES	Sample: SEZAINST(SERVICES). Required. Searched for in the first file found below: - /etc/services - userid.ETC.SERVICES (interactive) OR jobname.ETC.SERVICES (batch) - *hlq*.ETC.SERVICES
OE SNMP Server	Procedure names:	OSNMPD	
	Samples:	SEZAINST(OSNMPDPR)	
	Purpose:	Start the OE Simple Network Management Protocol agent.	
	Parameters:	Passed in the following configuration data sets:	
		hlq.PW.SRC	No sample provided. Required, cannot be explicitly allocated.
		hlq.SNMPTRAP.DEST	No sample provided. Optional, cannot be explicitly allocated.
		hlq.OSNMPD.DATA	Sample: /usr/lpp/tcpip/samples/osnmpd.data
	The SNMP agent uses some statements from TCPIP.DATA. The SNMP subagent uses some statements in the TCPIP.PROFILE.		
TCPIP	Procedure name:	TCPIP	
	Sample:	SEZAINST(TCPOPROC)	
	Purpose:	Starts the TCPIP address space.	
	Parameters:	Passed in the following configuration data sets:	
		PROFILE.TCPIP	Sample: SEZAINST(SAMOPROF). Required, can be dynamically or explicitly allocated. Recommend you explicitly allocate with //PROFILE DD.
		TCPIP.DATA	Sample: SEZAINST(TCPDATA). Required, can be dynamically or explicitly allocated. Searched for in the first file found below: - Environment variable RESOLVER_CONFIG - /etc/resolv.conf.

Configuration Data Sets

The following table lists the configuration data sets used by the TCP/IP servers and functions. It includes the name of the sample and the usage of the data set.

Table 4. TCP/IP Configuration Data Sets

Data Set/Search Order	Copied From	Usage
ETC.PROTO 1. /etc/protocol 2. userid/jobname.ETC.PROTO 3. *hlq*.ETC.PROTO	usr/lpp/tcpip/samples/protocol	Used to map types of protocol to integer values to determine the availability of the specified protocol. Required by several OS/390 TCP/IP OpenEdition components.
ETC.RPC	SEZAINST(ETCRPC)	Defines RPC applications to the Portmapper function.
ETC.SERVICES 1. /etc/services 2. userid/jobname.ETC.SERVICES 3. *hlq*.ETC.SERVICES	usr/lpp/tcpip/samples/services	Establishes port numbers for servers using TCP and UDP. Required for OE SNMP and OE RouteD.
FTP.DATA 1. //SYSFTPD 2. *userid/jobname*.FTP.DATA 3. SYS1.TCPPARMS(FTPDATA) 4. *hlq*.FTP.DATA	SEZAINST(FTCDATA) for the client and (FTPSDATA) for the server	Overrides default FTP client and server parameters for the FTP server.
HOSTS.LOCAL (or /etc/hosts)	SEZAINST(HOSTS)	Contains host names and IP addresses, used for non-DNS name resolution.
PROFILE.TCPIP 1. //PROFILE 2. *job_name.node_name*.TCPIP 3. *hlq .node_name*.TCPIP 4. *job_name*.PROFILE.TCPIP 5. *hlq*.PROFILE.TCPIP	SEZAINST(SAMOPROF)	Provides TCP/IP initialization parameters and specifications for network interfaces and routing.
TCPIP.DATA (or /etc/resolv.conf) 1. Environment variable RESOLVER_CONFIG 2. /etc/resolv.conf 3. //SYSTCPD 4. *userid*.TCPIP.DATA 5. *jobname*.TCPIP.DATA 6. SYS1.TCPPARMS(TCPDATA) 7. *hlq* .TCPIP.DATA	SEZAINST(TCPDATA)	Provides parameters for TCP/IP client programs.

Considerations for Multiple Instances of TCP/IP

For help in configuring multiple instances of TCP/IP, refer to the OpenEdition manuals. In particular, read the chapter on “setting up multiple transport drivers for TCP/IP” in *OS/390 OpenEdition Planning*. Also see the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

The information provided in this section is specific to running multiple instances of OS/390 TCP/IP OpenEdition. When running TCP/IP V3R2 for MVS and OS/390

TCP/IP OpenEdition under OMVS, it is suggested that TCP/IP for MVS be configured as the DEFAULT CINET physical file system. Configure the TCP/IP for MVS instance as outlined in its configuration guide. Use the information provided here when configuring an instance of OS/390 TCP/IP OpenEdition. The DPI subagent provided with OS/390 TCP/IP OpenEdition, OE routed, OE netstat, and OE snmpd are structured to run only with an instance of OS/390 TCP/IP OpenEdition.

Care must be taken when configuring multiple TCP/IP instances as OpenEdition CINET physical file systems. In particular, a TCP/IP instance and a certain set of its applications must pick up two TCPIP.DATA statements correctly:

1. HOSTNAME

   When the TCP/IP instance is started, it searches for its host name using the OE service __iphost(). Each TCP/IP instance must have its own hostname.

2. TCPIPJOBNAME

   Applications that must be associated with a particular instance must be able to issue the OE socket call setibmopt() to insure that they are associated with the right instance. This class of applications issues a __iptcpn() OE call to retrieve the TCPIPJOBNAME needed to issue the setibmopt() call on.

Both __iphost() and __iptcpn() follow the same search path to locate the file that should be searched:

1. ENVIRONMENT VARIABLE "RESOLVER_CONFIG=file/dataset"
2. /etc/resolv.conf
3. //SYSTCPD DD
4. *userid*.TCPIP.DATA or *jobname*.TCPIP.DATA
5. SYS1.TCPPARMS(TCPDATA)
6. *hlq*.TCPIP.DATA OR TCPIP.TCPIP.DATA

The OE services __iptcpn() and __iphost() use the above search path looking for an existing file. Once a file is located it is searched for the corresponding TCPIP.DATA statement. __iphost() looks for the HOSTNAME statement and __iptcpn() looks for the TCPIPJOBNAME statement. They do not search past the first file found.

In a single OS/390 TCP/IP OpenEdition instance environment, the HFS file /etc/resolv.conf can safely be used. In a multiple OS/390 TCP/IP OpenEdition instance environment, the environment variable RESOLVER_CONFIG should be used instead of the SYSTCPD DD, since SYSTCPD is searched only if /etc/resolv.conf doesn't exist. The environment variable can be set in a procedure by use of the ENVAR parameter in its parameter list. The following is an example of part of a procedure to start the SNMP Agent osnmpd:

```
//OSNMPD PROC
//OSNMPD EXEC PGM=EZASNMPD,REGION=4096K,TIME=NOLIMIT,
//      PARM=('POSIX(ON) ALL31(ON)',
//      ENVAR("RESOLVER_CONFIG=/etc/tcpv33a.data"),
//      '/ -c public')
```

When a TCP/IP instance is started, it determines its hostname by calling the __iphost(). If you are running multiple TCP/IP instances under OE, or a single instance, and GETHOSTNAME is returning the wrong name, check the search path

used by the PROC used to start TCP/IP to insure that the HOSTNAME statement is correct.

If you are having problems with either the subagent, snmp agent, orouted, or onetstat establishing an affinity with its TCP/IP, look at the message that is generated:

- Subagent

  The subagent terminates itself if its setibmopt() call failed. The following message is generated:

  ```
  EZZ3215I SNMP SUBAGENT: UNABLE TO CONNECT TO tcpipjobname
           (errno/errno2)
  ```

  In the TCPIP.DATA file that is first in the search path, set TCPIPjobname to the tcpipjobname. Check the BPXPRMxx member that was used to initialize OpenEdition to insure that the corresponding subfilesystem has the same name as the TCP/IP jobname. TCP/IP will have to be stopped and restarted for the subagent to retry the setibmopt() call.

- For onetstat, orouted, and osnmpd, there are two different types of messages that can be generated that indicate a problem in this area. The first type is when the TCPIPjobname could not be determined:

  ```
  EZZ6205I OE SNMP agent: Could not determine
           TCPIPjobname, using default of 'INET'
  EZZ4986I OE RouteD could not determine TCPIPjobname,
           using default of 'INET'
  EZZ2376I Could not determine TCPIPjobname,
           using default of 'INET'
  ```

  If you have INET installed and get this message, the setibmopt will work. If you have CINET installed and get this message, the setibmopt() call will most likely fail. Add or correct the TCPIPjobname statement in the OE TCPIP.DATA search path. The following message is generated when this occurs:

  ```
  EZZ6272I OE SNMP agent: Could not establish affinity with
           with tcpipjobname (errno/errno2)
  EZZ4987I OE RouteD could not establish affinity
           with tcpipjobname (errno/errno2)
  EZZ2377I Could not establish affinity with tcpipjobname
           (errno/errno2) - can not provide the requested
           information
  ```

The following list describes the general steps that should be followed when configuring multiple instances of TCP/IP:

1. Create another started task (TCPOPROC) for each TCPIP address space. Do not use the same proc name for the TCPOPROC started task for each instance of TCP/IP running on the same MVS. If you use the same proc name for multiple TCPOPROC started tasks and use dot qualification for distinction between task names, unpredicatible results can occur when trying to connect to a specific TCP/IP. For more information on running with multiple TCP/IPs, see *OS/390 OpenEdition Planning*. Also see the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

   - Add or modify the //PROFILE statement to point to the correct PROFILE.TCPIP data set to be used by this instance of TCP/IP.

- Add or modify the //SYSTCPD statement to point to the correct TCPIP.DATA data set, if /etc/resolv.conf does not exist. Otherwise, set the ENVAR parameter in parmlist to point to the correct TCPIP.DATA file.

2. Create another PROFILE.TCPIP data set
   - To use a different high-level qualifier for dynamically allocated data sets, modify the DATASETPREFIX parameter.
   - Modify the parameters as if this data set were on another MVS system. For example, you might:
     - Give the HOME statement different IPADDRS values
     - Specify different DEVICE and LINK statements
3. Insure that the BPXPRMxx member (found in SYS1.PARMLIB) used to start OMVS is configured to start common INET sockets processing:

   ```
   FILESYSTYPE TYPE(CINET)
               ENTRYPOINT(BPXTCINT)
   ```

   Insure that the TCPIPJOBNAME associated with each instance of TCP/IP has been configured as an AF_INET physical file system:

   ```
   SUBFILESYSTYPE NAME(tcpipjobname)
                  TYPE(CINET)
                  ENTRYPOINT(EZBPFINI)
   ```

   The instance of TCP/IP that is always started should be designated as the DEFAULT by adding DEFAULT after the ENTRYPOINT definition. OMVS does not perform correctly with respect to AF_INET socket usage if the DEFAULT AF_INET physical file system is not started.

   Check to insure that the INADDRANYPORT assignment does not conflict with PORT assignments in the PROFILE.TCPIP data sets associated with each instance of TCP/IP:

   ```
   NETWORK DOMAINNAME(AF_INET) DOMAINNUMBER(2) MAXSOCKETS(10000)
           TYPE(CINET) INADDRANYPORT(4901) INADDRANYCOUNT(100)
   ```

4. Create a TCPIP.DATA file for each instance of TCP/IP:
   - To use a different high-level qualifier for dynamically allocated data sets, modify the DATASETPREFIX parameter.
   - Modify HOSTNAME to identify this instance of TCP/IP.
   - Modify TCPIPJOBNAME to contain the name of the address space that is started for this instance of TCP/IP.
5. If you plan on running multiple instances of the OE FTP server, create separate FTP.DATA files. For more information on OE FTP, see Chapter 7, “Configuring the OE File Transfer Protocol (FTP) Server” on page 133. The OE FTP server does not form an affinity with a particular instance of TCP/IP. One FTP server can service clients from multiple instances of TCP/IP. This is also true for the OE Telnet server and all OE applications not explicitly referenced in this section.
6. Create a separate procedure for each instance of OE routed and OE snmpd. These servers must be associated with a single instance of TCP/IP. In the procedure that is used to start them, use the ENVAR parameter as described previously to point to the TCPIP.DATA file associated with the correct instance of TCP/IP.

Administration Overview

After your TCP/IP system is configured, you can use these MVS commands to dynamically start, stop, and control the servers:

- START
- STOP
- DISPLAY TCPIP
- VARY TCPIP

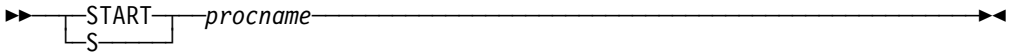
All of the servers or address spaces except PORTCPRC support START and STOP. PORTCPRC supports START but does not support STOP.

Recommendation: Although the MVS commands can accept *procname.identifier* to specify the server or address space, it is preferred that you use the member name of the cataloged procedure on the PORT statement in *hlq*.PROFILE.TCPIP and on all these MVS commands.

Starting and Stopping TCP/IP Servers

START Command

Use the START command to dynamically start a TCP/IP server or address space (including the TCP/IP address space).



procname
The name of a member in a cataloged procedure library. For the servers, this should be the same name specified on the PORT statement in the PROFILE.TCPIP data set.

STOP Command

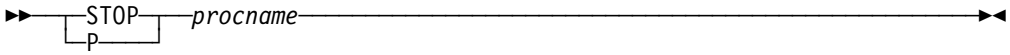
Use the STOP command to stop a TCP/IP server or address space (including the TCP/IP address space) that is in execution.

When you issue STOP TCPIP, the following sequence of events occurs, depending on whether connected servers have outstanding calls to TCP/IP.

For each server with outstanding calls to TCP/IP: The TCP/IP address space notifies the server that TCP/IP is coming down and requests that the server terminate normally.

If the server does not terminate normally, TCP/IP causes the server to abend with abend code 422. The abend does not appear in a dump; however it is recorded in the SYS1.LOGREC data set. The outstanding socket call receives error number 1041 EIBMBADPOSTCODE.

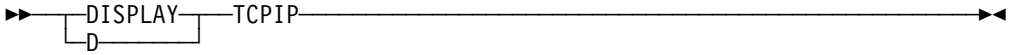
For each connected server that does not have outstanding calls: The TCP/IP address space notifies the server that TCP/IP is coming down and drives the server's asynchronous error exit routine, if there is one.



procname
The name of the procedure you want to stop. This should be the same member name used to start the server on the START command.

Using the DISPLAY TCPIP Command

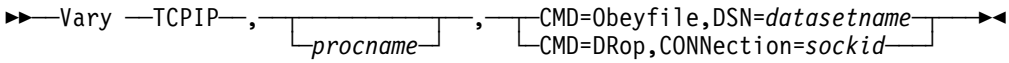
Use the DISPLAY TCPIP command to display the status of the current TCP/IP images.



Using the VARY TCPIP Command

Use the VARY TCPIP command to control some functions of the TCP/IP address space from the operator's console.

This is the general format of the VARY TCPIP command:



For more information on the VARY TCPIP command, see Chapter 3, “Configuring the TCPIP Address Space” on page 33.

Chapter 3. Configuring the TCPIP Address Space

Before You Configure...:

Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

When TCPIP is started, it loads configuration information from a profile data set. This data set is referred to as *hlq*.PROFILE.TCPIP or PROFILE.TCPIP in this book. This chapter describes how to configure the TCPIP address space by updating the configuration statements in this data set. It also describes how you can dynamically change the configuration for the current session with the VARY TCPIP command.

Configuration Process

Steps to configure the TCPIP address space:

1. Update the TCPIP cataloged procedure
2. Specify configuration statements in PROFILE.TCPIP

Step 1: Update the TCPIP Cataloged Procedure

Copy the TCPIP cataloged procedure in *hlq*.SEZAINST(TCPOPROC) to your system or recognized PROCLIB and modify it to suit your local conditions. Specify TCPIP parameters and remove or change the DD statements as required. The started task user ID of the TCP/IP system address space must match the NAME parameter on the SUBFILESYSTYPE statement in the BPXPRMxx member of 'SYS1.PARMLIB' used to start OE. For more information on BPXPRMxx, see the *OS/390 OpenEdition File System Interface Reference*.

TCPIP Cataloged Procedure (TCPOPROC)

```
//TCPIP33X  PROC PARMS='CTRACE(CTIEZB00)'
//*
//* OS/390 TCP/IP OpenEdition
//* SMP/E Distribution Name: EZBOPROC
//*
//*     5645-001 5655-HAL (C) Copyright IBM Corp. 1989, 1997.
//*     All rights reserved.
//*     US Government Users Restricted Rights -
//*     Use, duplication or disclosure restricted
//*     by GSA ADP Schedule Contract with IBM Corp.
//*     See IBM Copyright Instructions
//*
//TCPIP    EXEC PGM=EZBTCPIP,
//            PARM='&PARMS',
//            REGION=7500K,TIME=1440
//*
//*     The C runtime libraries should be in the system's link list
```

```
//*    or add them to the STEPLIB definition here.  If you add
//*    them to STEPLIB, they must be APF authorized.
//*
//*STEPLIB  DD...
//*
//*    SYSPRINT contains run-time diagnostics from TCPIP. It may be
//*      a data set or SYSOUT.
//*    SYSERROR contains error messages from TCPIP that occurred
//*      while processing the PROFILE.
//*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
//SYSOUT   DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=137,BLKSIZE=137)
//SYSERROR DD SYSOUT=*
//*
//*    TCPIP reads the parameters from a data set with name
//*    TCPIP.nodename.TCPIP or with name TCPIP.PROFILE.TCPIP.
//*    See the chapter on "Configuring the TCPIP Address Space" in
//*    the Configuration Guide for more information. A sample of
//*    such a profile is included in member SAMOPROF of the
//*    SEZAINST data set.
//*
//PROFILE  DD DISP=SHR,DSN=TCPIP.PROFILE.TCPIP
//*
//*    SYSTCPD explicitly identifies which data set is to be
//*      used to obtain the parameters defined by TCPIP.DATA.
//*      The SYSTCPD DD statement should be placed in the TSO logon
//*      procedure or in the JCL of any client or server executed
//*      as a background task.  The data set can be any sequential
//*      data set or a member of a partitioned data set (PDS).
//*
//*    For more information please see "Understanding TCP/IP Data
//*    Set Names" in the Configuration Guide.
//*
//SYSTCPD  DD DSN=TCPIP.SEZAINST(TCPDATA),DISP=SHR
```

Using Output Data Sets

In the TCPIP address space, the SYSPRINT and SYSERROR data sets defined with a DD statement must have a variable blocked (VB) format and a logical record length (LRECL) of 137. You can allocate these as partitioned or sequential data sets, but be aware that partitioned data sets cannot be reused if they have filled or if the members already exist.

Output to these data sets is handled in the following manner:

- If only a primary data set is defined, the primary data set is overwritten.
- If any of the alternate data sets are specified, initially the primary data set is overwritten. Should this data set fill, it is closed and output is sent to the next alternate data set. Existing data in the alternate data sets is overwritten. This process continues until all defined data sets have been filled. Once this occurs, the primary data set is overwritten again and the cycle continues.

Specifying the CTRACE Keyword

Specify the CTRACE parameter on the PARMS=CTRACE(CTIEZB00) keyword of the EXEC JCL statement in the TCP/IP started procedure to indicate the member of the SYS1.PARMLIB to be used by TCP/IP Component Trace processing. A sample of this parmlib member can be found in *hlq*.SEZAINST. Specify the ENVAR parameter on the PARMS=CTRACE(CTIEZB00) keyword to override the resolver file.

For more information on setting the environment variable RESOLVER_CONFIG using the ENVAR parameter, see "Considerations for Multiple Instances of TCP/IP" on page 26. You can find a description of the MVS Component Trace support in TCP/IP in *OS/390 TCP/IP OpenEdition Diagnosis Guide*.

Recommendations: To ensure that no security-sensitive data is exposed after a dump has occurred, you might want to use SYSMDUMP data sets. For more information on dump processing, see the *OS/390 TCP/IP OpenEdition Diagnosis Guide*.

Step 2: Specify Configuration Statements in PROFILE.TCPIP

During initialization of the TCPIP address space, system operation and configuration parameters are read from a configuration profile data set.

A sample of this data set, SEZAINST(SAMOPROF), is shown in "Sample Profile Configuration Data Set (SAMOPROF)" on page 41. You can copy and modify this sample to use as your default configuration profile.

As you create and name your configuration profile, be aware of the search order TCP/IP uses to find this data set:

- //PROFILE DD DSN=

  TCP/IP looks first for a data set specified by the //PROFILE DD statement in the cataloged procedure.

- *job_name.node_name*.TCPIP

  If there is no //PROFILE statement, TCP/IP looks next for *job_name.node_name*.TCPIP, where *node_name* is the node name returned from __ipnode() (the node name specified on the VMCF initialization record). For more information on __ipnode(), see the *OS/390 C/C++ Run-Time Library Reference*.

- If this data set is not found, the program uses the first of the following data sets it finds:

  – *hlq.node_name*.TCPIP
  – *job_name*.PROFILE.TCPIP
  – *hlq*.PROFILE.TCPIP

To customize your system, specify system operation parameters and network configuration information in this data set using the configuration statements listed in Table 6 on page 39. You can find the complete statement syntax and descriptions in alphabetical order in "PROFILE.TCPIP Configuration Statements" on page 45.

You can also put many of these statements in a separate data set, process it with the VARY TCPIP command, and dynamically change the TCP/IP configuration

established by the PROFILE.TCPIP data set. For more information, see “VARY Command—TCPIP Address Space” on page 96.

DEVICE and LINK Statements

OS/390 TCP/IP OpenEdition allows a single TCPIP address space to drive multiple instances of any supported device. To configure your devices, add the appropriate DEVICE and LINK statements to the configuration data set. The LINK statements show how to define a network interface link associated with the device and are included with the DEVICE statement for that device type.

Because devices are not automatically initialized, you must also specify a START statement in the configuration data set to start each device automatically.

There are DEVICE and LINK statements to configure the following:

- LCS devices
- CTC devices
- ATM devices (for network management data retrieval only)
- Virtual IP address (VIPA)
- CLAW devices (for example, channel-attached RISC System/6000)

You can add new DEVICE and LINK statements using the VARY TCPIP command. You can also delete and redefine existing statements.

When you add new LINK statements, any corresponding GATEWAY, HOME, and TRANSLATE statements coded to include the new links are treated as replacements for active statements. Therefore, when you code the GATEWAY, HOME, or TRANSLATE statements in the OBEYFILE data set, be sure to include existing links that you want to have active in your configuration.

For more information on VARY TCPIP, see “VARY Command—TCPIP Address Space” on page 96.

Routing Statements

TCP/IP supports static and dynamic routing. You can define static routes in the IP host using the GATEWAY statement in *hlq*.PROFILE.TCPIP. You can also configure ORouteD to implement the Routing Information Protocol (RIP) for dynamic routing. The use of the GATEWAY statement and BSDROUTINGPARMS statement in PROFILE.TCPIP differs for each of these two methods. There are some general rules you can follow when configuring ORouteD.

ORouteD

- Use BSDROUTINGPARMS in PROFILE.TCPIP to define your routing parameters.
- The RouteD server uses the following search order to locate the GATEWAYS configuration data set or file. Only the first file in the search order that can be opened is read to determine the gateway statements.
  1. If the environment variable GATEWAYS_FILE has been defined, ORouteD uses this value as the name of an MVS data set or HFS file to access the gateways file. The syntax for an MVS data set name is //'*mvs.dataset.name*'. The syntax for an HFS file name is /*directory/subdirectory/file.name*.

2. /etc/gateways
3. *hlq*.ETC.GATEWAYS

See Chapter 10, “Configuring the OE RouteD Server” on page 219 for details.

Primary or Alternate Network Attachments Support

TCP/IP supports multiple attachments and IP addresses on the same LAN, providing redundant paths to other hosts or routers on directly-attached LANs. When multiple attachments to the same LAN are configured, one attachment is the primary path to hosts and routers on that LAN, and others are secondary.

The primary path to hosts and routers on the LAN is defined in one of two ways:

- The primary path can be specified in the PRIMARYINTERFACE statement.
- If no PRIMARYINTERFACE statement is configured, the first link for each defined subnetwork in the list of HOME addresses is primary.

All other interfaces on a directly-attached LAN are secondary.

TCP/IP uses the primary interface for all outbound traffic to hosts and routers on directly-attached LANs, as long as the primary interface is functioning. If a primary interface fails, outbound traffic is sent on a secondary interface.

Inbound traffic is not affected, because assignment of paths into MVS is done by routing protocols out in the network.

Using Virtual IP Addressing Support

The purpose of Virtual IP Addressing (VIPA) is to free other TCP/IP hosts from dependence on particular network attachments to MVS. Prior to VIPA, other hosts got bound to one of MVS TCP/IP's home IP addresses and, therefore, to a particular network attachment (for example, a controller or adapter) to MVS. VIPA provides an IP address that selects a TCP/IP image (and MVS system if there is only one image on an MVS system) without selecting a specific network attachment. Other hosts that connect to MVS TCP/IP applications can send data to an MVS VIPA via whatever paths are selected by the routing protocols. VIPA provides tolerance of failures of MVS network attachment hardware.

VIPA uses a virtual device and a virtual IP address. The virtual device will always be active and never see a failure. A virtual IP address will be the home address for the virtual device, but there will be no physical interface associated with it. Inbound packets that have the virtual IP address as the destination can be routed through any one of the real physical interfaces to MVS. Failure of a real MVS network interface is handled by routing inbound traffic to another interface using RIP provided by the ORouteD application.

See “DEVICE and LINK Statement—Virtual Devices (VIPA)” on page 64 for more information on defining virtual addresses using DEVICE and LINK statements. See “BSDROUTINGPARMS Statement” on page 48 for more information on specifying virtual links on the BSDROUTINGPARMS statement. See “HOME Statement” on page 71 for more information on specifying virtual links on the HOME statement.

Summary of TCPIP Configuration Statements

Table 5 lists the TCP/IP configuration statements and the minimum, maximum, and default parameter values. Following Table 5, Table 6 on page 39 contains a brief description of each configuration statement, along with the page where further information can be found. Complete descriptions of the configuration statements follow the tables.

Table 5 (Page 1 of 2). TCPIP Configuration Statement Parameters — Min, Max, and Default Values

STATEMENT **parameter**	**Minimum**	**Maximum**	**Default**
ARPAGE (in minutes)	1	1440	20
BSDROUTINGPARMS			
mtu (size in bytes)	1	65535	DEFAULTSIZE (576)
metric	0	14	0
DEVICE/LINK ATM			
ifspeed (in bits per second)	0	2147483647	0
DEVICE/LINK CTC			
iobuffersize[1]	32K	65535	32K
ifspeed (in bits per second)	0	2147483647	4500000
DEVICE/LINK LCS			
ifspeed (in bits per second)	0	2147483647	4000000
DEVICE/LINK CLAW			
read_buffers	1	2147483647	20
write_buffers	1	2147483647	20
read_size[2]	1024	4096	4096
write_size[2]	1024	4096	4096
ifspeed (in bits per second)	0	2147483647	100000000
GATEWAY			
max_packet_size (in bytes)	1	65535	DEFAULTSIZE (576)
IPCONFIG			
arpto (in seconds)	60	86400	1200
reassemblytimeout (in seconds)	1	240	60
ttl	1	255	64
KEEPALIVEOPTIONS			
interval[3] (in minutes)	0	35,791	120
PKTTRACE			
abbrev	1	65535	200
prot	0	255	

Table 5 (Page 2 of 2). TCPIP Configuration Statement Parameters — Min, Max, and Default Values

STATEMENT parameter	Minimum	Maximum	Default
port_num	1	65535	
PORT			
port_num	1	65535	
PORTRANGE			
starting_port_num	1	65535	
num_ports	1	65535	
SACONFIG			
osasf_port_number	0	65535	
agent_port_number	1	65535	161
SOMAXCONN			
maximum_queue_depth	1	2147483647	10
TCPCONFIG			
default_keepalive_interval (in minutes)	0	35791	120
tcp_send_buffer_size	256	256K	16384 (16K)[4]
tcp_receive_buffer_size	256	256K	16384 (16K)[4]
TRUNC			
line_length	64	255	255
UDPCONFIG			
udp_send_buffer_size	1	65535	65535
udp_receive_buffer_size	1	65535	65535

Notes:

1. The only valid values are 32K, 32768, and 65535.
2. These values must be a multiple of 1K.
3. KEEPALIVEOPTIONS interval allows zero (0) to turn it off. The maximum value of 35,791 is approximately 2**31-1 milliseconds.
4. The default for TCPSENDBFRSIZE and TCPRCVBUFRSIZE is approximately 16384 (16K), but will change slightly with service changes.

Table 6 contains a brief description of each configuration statement, along with the page where further information can be found.

Table 6 (Page 1 of 3). Summary of TCPIP Address Space Configuration Statements

Statement	Description	Page
ARPAGE	Alters the number of minutes before an ARP table entry is deleted.	45

Table 6 (Page 2 of 3). Summary of TCPIP Address Space Configuration Statements

Statement	Description	Page
ASSORTEDPARMS	Passes initialization parameters to TCPIP.	46
BSDROUTINGPARMS	Defines network interface information. Used by the ORouteD server.	48
DATASETPREFIX	Specifies a prefix that will be used instead of the default high-level qualifier for the dynamic allocation of data sets and in the hierarchical search sequence.	50
DELETE	Removes a device, link, port, or portrange.	51
DEVICE and LINK	ATM Devices	53
DEVICE and LINK	CTC devices	54
DEVICE and LINK	LCS devices	56
DEVICE and LINK	CLAW devices	61
DEVICE and LINK	Virtual Devices	64
GATEWAY	Defines IP routing table entries for static routes.	65
HOME	Provides a list of home addresses and associated link names.	71
INCLUDE	Causes another data set that contains profile configuration statements to be included at this point.	73
IPCONFIG	Specifies IP configuration values.	75
ITRACE	Controls tracing for configuration.	76
KEEPALIVEOPTIONS	Specifies the operating parameters of the TCP keep-alive mechanism.	78
LINK	Defines network interface links.	36
PKTTRACE	Defines the conditions used to select IP packets as candidates for tracing and subsequent analysis.	79
PORT	Reserves a port for one or more given process names.	83
PORTRANGE	Reserves a range of ports for one or more process names.	85
PRIMARYINTERFACE	Specifies which link is to be considered the primary interface.	87
SACONFIG	Specified SNMP subagent parameters.	89
SOMAXCONN	Specifies a maximum connection length for the connection request queues created by the socket call listen().	90
START	Starts the specified device.	91
STOP	Stops the specified device.	92

Table 6 (Page 3 of 3). Summary of TCPIP Address Space Configuration Statements

Statement	Description	Page
TCPCONFIG	Specifies TCP parameters.	93
TRANSLATE	Indicates the relationship between an internet address and the network address.	94
TRUNC	Used to allow sequence numbers in profile data set.	95
UDPCONFIG	Specifies UDP parameters.	95

Sample Profile Configuration Data Set (SAMOPROF)

The following sample configuration data set is provided in *hlq*.SEZAINST(SAMOPROF). This member is used to configure the TCPIP address space.

```
;
; TCPIP.PROFILE.TCPIP
; ===================
;
; COPYRIGHT = NONE.
;
; This is a sample configuration file for the OE TCPIP address space.
;
; NOTES:
;
;    The device configuration statements MUST be changed to match your
;    hardware and software configuration.
;
;
; For more information about this file, see "Configuring the TCPIP
; Address Space" and "Configuring the OE Telnet Server" in the
; Configuration Guide.
;
; ------------------------------------------------------------------------
;
;
; ------------------------------------------------------------------------
;
; Flush the ARP tables every 20 minutes.
;

ARPAGE 20

;
; You can specify DATASETPREFIX in the PROFILE.TCPIP and
; TCPIP.DATA data sets. If this statement is used in a profile or
; configuration data set that is allocated to a client or a server, then
; that client or server dynamically allocates additional required data
; sets using the value specified for DATASETPREFIX as the data set name
; prefix.  The DATASETPREFIX parameter can be up to 26 characters long,
```

```
; and the parameter must NOT end with a period.
;
; For more information please see "Understanding TCP/IP Data Set
; Names" in the Customization and Administration Guide.
;
DATASETPREFIX TCPIP
;

;
; ------------------------------------------------------------------
;
; Reserve low ports for servers
;
TCPCONFIG        RESTRICTLOWPORTS
UDPCONFIG        RESTRICTLOWPORTS

;
; ------------------------------------------------------------------
;
; Hardware definitions:
;

;
; LCS1 is a 3172 Model 1 with a Token-Ring and Ethernet adapter.
;

;
; To use these device and link statements, update the statements to
; reflect your installation configuration and remove the semicolon
;DEVICE LCS1   LCS           BA0
;LINK TR1   IBMTR    0 LCS1
;LINK ETH1  ETHERNET 1 LCS1

;
; LCS2 is a 3172 Model 2 with a FDDI adapter.
;

;
; To use these device and link statements, update the statements to
; reflect your installation configuration and remove the semicolon
;DEVICE LCS2   LCS           BE0
;LINK FDDI1 FDDI     0 LCS2

;
; ------------------------------------------------------------------
;
; HOME Internet (IP) addresses of each link in the host.
;

HOME
; To use this home statement, update the ipaddress and linknames
; to reflect your installation configuration and remove the semicolon
;  130.50.75.1     TR1
```

```
;  193.5.2.1       ETH1
;  9.67.43.110     FDDI1

;
; ----------------------------------------------------------------------
;
; The      PRIMARYINTERFACE statement is used to specify which interface
; is the primary interface.
;
; If PRIMARYINTERFACE is not specified, then the first link in the HOME
; statement is the primary interface, as usual.
;

; To use this primary statement, update the and linkname
; to reflect your installation configuration and remove the semicolon
; PRIMARYINTERFACE TR1

;
; ----------------------------------------------------------------------
;
; IP routing information for the host.  All static IP routes should
; be added here.
;

GATEWAY
; To use this GATEWAY statement, update the addresses and links
; to reflect your installation configuration and remove the semicolon
;
; Direct Routes - Routes that are directly connected to my interfaces.
;
; Network  First Hop  Link Name Packet Size  Subnet Mask  Subnet Value

; 130.50       =         TR1       2000       0.0.255.0    0.0.10.0
; 193.5.2      =         ETH1      1500       0
; 9            =         FDDI1     4000       0.255.255.0  0.67.43.0

;
; Indirect Routes - Routes that are reachable through routers on my
;                   network.
;
; Network  First Hop  Link Name Packet Size  Subnet Mask  Subnet Value

; 193.12.2 130.50.10.1   TR1       2000       0
; 10.5.6.4 193.5.2.10    ETH1      1500       HOST

;
;
; Default Route - All packets to an unknown destination are routed
;                 through this route.
;
; Network  First Hop  Link Name Packet Size  Subnet Mask  Subnet Value

; DEFAULTNET 9.67.43.1 FDDI1    DEFAULTSIZE  0
```

```
;
; ---------------------------------------------------------------------
;
; orouted Routing Information
;
; if you are using orouted, comment out the GATEWAY statement and
; update the BSDROUTINGPARMS statement to reflect your installation
; configuration and remove the semicolon
; ;  Link     Maxmtu    Metric    Subnet Mask      Dest Addr
; BSDROUTINGPARMS false
;    TR1        2000        0      255.255.255.0    0
;    ETH1       1500        0      255.255.255.0    0
;    FDDI1  DEFAULTSIZE     0      255.255.255.0    0
; ENDBSDROUTINGPARMS
;

;
; ---------------------------------------------------------------------
;
; Use TRANSLATE to specify the hardware address of a specific IP
; address.  See the Customization and Administration Guide for more
; information.
;

TRANSLATE
; A null translate statement issues the warning message EZZ0323I

;
; ---------------------------------------------------------------------
;
; Turn off all tracing.  If tracing is to be used, change the following
; line.  To trace the configuration component, for example, change
; the line to ITRACE ON CONFIG 1

ITRACE OFF
;
; ---------------------------------------------------------------------
; The ASSORTEDPARMS NOFWD will prevent the forwarding of IP packets
; between different networks.  If NOFWD is not specified, IP packets
; will be forwarded between networks when this host is a gateway.
;
; Even though RESTRICTLOWPORTS was specified on TCPCONFIG and UDPCONFIG,
; ASSORTEDPARMS default would have been to reset RESTRICTLOWPORTS to off
; So it is respecified here.
; If the TCPCONFIG and UDPCONFIG followed ASSORTEDPARMS, RESTRICTLOWPORT
; would not have to be done twice.

ASSORTEDPARMS
  NOFWD
  RESTRICTLOWPORTS
ENDASSORTEDPARMS
; NOFWD            issues the informational message EZZ0334I
; RESTRICTLOWPORTS issues the informational message EZZ0338I
```

```
;
; ---------------------------------------------------------------------
;
; Start all the defined devices.
;

; To use these START statements, update the device name
; to reflect your installation configuration and remove the semicolon
; START LCS1
; START LCS2
```

PROFILE.TCPIP Configuration Statements

This section contains the configuration statements for the *hlq*.PROFILE.TCPIP configuration data set.

Statement Syntax

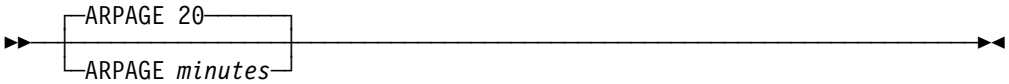
Statement syntax is the same in both the configuration data set (*hlq*.PROFILE.TCPIP) and the VARY TCPIP,,CMD=OBEYFILE data set. The following formatting restrictions apply to configuration statements:

- Entries in a configuration data set are free format; blanks, comments, and end-of-record are ignored.
- A configuration statement consists of a statement name followed by a required blank, and usually one or more positional arguments. Separate each argument by one or more blanks or end-of-record.
- An argument followed by a comment must have a blank before the semicolon.
- A semicolon begins a comment. Comments act as blanks, separating words without affecting their meaning.
- Statements can be split across multiple lines.
- Sequence numbers are not allowed unless using the TRUNC statement.
- Lowercase letters are translated to uppercase before the statements are executed.
- An END*statement* terminates a number of statements, such as ASSORTEDPARMS. If the END*statement* is omitted, all subsequent tokens in the data set are interpreted as parameters for that configuration statement.
- If a syntax error is encountered in a list of parameters, such as a HOME list, the rest of the entries in the list are ignored.

ARPAGE Statement

Use the ARPAGE statement to change the number of minutes between creation or revalidation, and deletion. By default, TCPIP deletes ARP table entries 20 minutes after creation or revalidation. An ARP table entry is revalidated when another ARP packet is received from the same host specifying the same hardware address.

Syntax



Parameters

minutes

The number of minutes between creation or revalidation of an ARP table entry and deletion of the entry.

This number is an integer in the range of 1 through 1440 (24 hours). The default is 20 minutes.

Examples

This example clears the ARP tables every 10 minutes.

```
ARPAGE 10
```

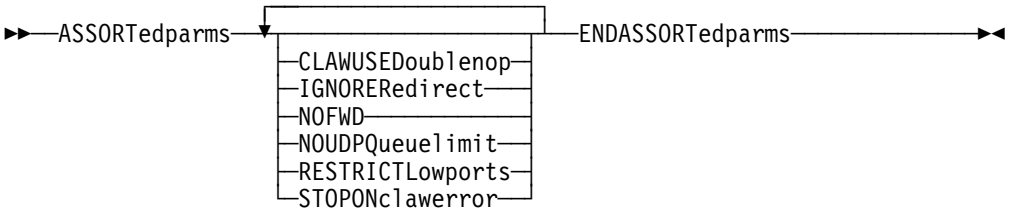
Usage Notes

IPCONFIG ARPTO allows you to specify the number of *seconds* between creation or revalidation and deletion.

ASSORTEDPARMS Statement

Use the ASSORTEDPARMS statement to pass initialization parameters to TCPIP.

Syntax



Parameters

CLAWUSEDOUBLENOP

Forces channel programs for Common Link Access to Workstation (CLAW) devices to have two NOP CCWs to end the channel programs. This is required for some vendor devices, and only applies to first-level MVS systems. The CLAWUSEDOUBLENOP parameter is confirmed by the message:

```
CLAWUSEDOUBLENOP is set
```

IGNOREREDIRECT

Causes TCPIP to ignore ICMP Redirect packets. The IGNOREREDIRECT parameter is confirmed by the message:

```
ICMP will ignore redirects
```

If you are using ORouteD, use this option because ORouteD does not support ICMP redirects.

NOFWD
Stops the transfer of data between networks by disabling IP datagram routing between different network interfaces. This statement can be used for security or to ensure correct usage of limited resources. The NOFWD parameter is confirmed by the message:

```
IP forwarding is disabled.
```

If either ASSORTEDPARMS NOFWD or IPCONFIG NODATAGRAMFWD is specified in a profile, or if neither the ASSORTEDPARMS nor the IPCONFIG statement is specified, forwarding is disabled. If the ASSORTEDPARMS or IPCONFIG statement is specified and the NOFWD and NODATAGRAMFWD parameters are not included, forwarding is enabled.

NOUDPQUEUELIMIT
Causes TCPIP to relax the default limit of 21 incoming datagrams queued on a UDP port.

The NOUDPQUEUELIMIT parameter is confirmed by the message:

```
No limit on incoming UDP datagram queue set
```

RESTRICTLOWPORTS
When set, ports 1 through 1023 are reserved for users of the PORT and PORTRANGE statement. The RESTRICTLOWPORTS parameter is confirmed by the messages:

```
UDP ports 1 thru 1023 are reserved
TCP ports 1 thru 1023 are reserved
```

STOPONCLAWERROR
Stops channel programs (HALTIO and HALTSIO) when a CLAW device error is detected. The STOPONCLAWERROR parameter is confirmed by the message:

```
STOPONCLAWERROR is enabled
```

Examples

This example shows the use of the NOFWD parameter on the ASSORTEDPARMS statement.

```
ASSORTEDPARMS
  NOFWD
ENDASSORTEDPARMS
```

Usage Notes

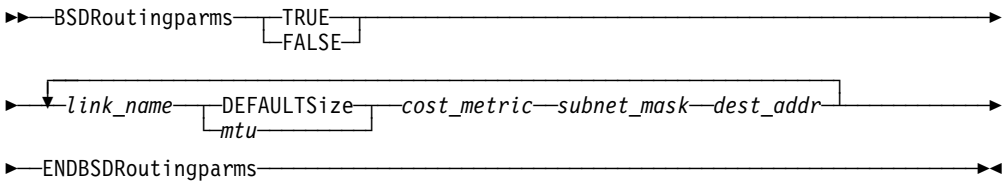
- The ENDASSORTEDPARMS statement is the delimiter for the ASSORTEDPARMS statement. If ENDASSORTEDPARMS is omitted, subsequent tokens are saved until the maximum string length is reached, and an error message is generated. The ASSORTEDPARMS string is then cleared, and processing resumes at the next token that is recognized as a valid configuration command.
- If you specify NOUDPQUEUELIMIT when you are running untested applications on your system, a malfunctioning application can tie up available storage.
- If some but not all of the ASSORTEDPARMS are specified, by default, those not specified are set to off.

- If any of the ASSORTEDPARMS are specified on other statements (IPCONFIG, TCPCONFIG, or UDPCONFIG), the settings from the last statement processed are used. For example, if RESTRICTLOWPORTS is not specified on ASSORTEDPARMS (and thus defaults to off) but is specified on a subsequent TCPCONFIG statement, RESTRICTLOWPORTS will be set for TCP.

BSDROUTINGPARMS Statement

Use the BSDROUTINGPARMS statement to define the characteristics of each link defined at the host over which ORouteD will send routing information to client routers.

Syntax



Parameters

TRUE
Specifies that the maximum packet size for the interface is always used, regardless of the final destination host.

FALSE
Specifies that the default maximum packet size of 576 is used when sending to networks that are not locally attached.

link_name
The name of the link as defined in a LINK statement. Each link should be defined once in the BSDROUTINGPARMS statement.

mtu
The maximum transmission unit (MTU) in bytes for the network or host. The special entry DEFAULTSIZE in this field requests that TCPIP supply a default value of 576. You can initially specify DEFAULTSIZE as the packet size for each network. Specify other values during later performance tuning. For virtual links, this field is meaningful only for point-to-point channels between TCPIP stacks in the same MVS image (for example, CTC).

cost_metric
The metric associated with the cost of use for the link. When sending routing information over this link, ORouteD will add a metric value to the routing metrics for the routes that are to be broadcast over this link. The metric value that is added will be the value specified in the BSDROUTINPARMS section incremented by one. If a metric of zero is specified, a metric value of one will be added, which is the default cost for a directly-connected network. If a metric of one is specified, a metric value of two will be added. The higher metric causes the route over this link to be less preferred. The range is from 0 to 14. A metric of 0 is usually coded so that the routes sent over the interface will be the most preferred.

subnet_mask
A bit mask (expressed in dotted-decimal form) defining the subnet mask associated with the link. The bits must be contiguous in the network portion of the *subnet_mask*. If the *subnet_mask* equals zero or the network class mask, RouteD will default the subnetwork mask to the network class mask.

dest_addr
Destination address applies to point-to-point links only. If the link is a point-to-point link, insert the address of the host on the other end of the link. If the interface connects to a normal network, use 0 as the destination address. For virtual links, this field should be 0. The CTC and CLAW link types are defined as point-to-point.

Examples

- This example shows the BSDROUTINGPARMS statement for several types of LAN media.

```
;  link     maxmtu   metric   subnet mask     dest addr
BSDROUTINGPARMS false
   TR1       2000       0     255.255.255.0   0
   ETH1      1500       0     255.255.255.0   0
   FDDI1  DEFAULTSIZE   0     255.255.255.0   0
ENDBSDROUTINGPARMS
```

- This example includes a link, `LINK3`, that is a point-to-point link between host MVS1 and host 128.84.54.6.

```
;
;  link     maxmtu    metric    subnet_mask    dest_addr
BSDROUTINGPARMS false
   LINK1  DEFAULTSIZE    0      255.255.255.0     0
   LINK2  DEFAULTSIZE    0      255.255.255.0     0
   LINK3  1500           0      255.255.255.0     128.84.54.6
ENDBSDROUTINGPARMS
```

- This example shows the definitions for virtual devices.

```
BSDROUTINGPARMS false
   VLINK1   DEFAULTSIZE  0  255.255.255.252  0
   VLINK2   DEFAULTSIZE  0  255.255.255.252  0
ENDBSDROUTINGPARMS
```

Usage Notes

- Use the BSDROUTINGPARMS statement whenever you are running the ORouteD server. If you are not running ORouteD, this statement is not relevant.
- The ORouteD server does not have to be restarted if you have *added* new links in the BSDROUTINGPARMS statement using the VARY TCPIP command. When issuing the VARY TCPIP command, include both the HOME and BSDROUTINGPARMS statements. Do not *change* BSDROUTINGPARMS for links already in use by ORouteD.
- For rules on defining virtual IP addresses for virtual links, see the “HOME Statement” on page 71.
- The maximum transmission unit (MTU) and metric of any other links with a destination address in the same subnet are updated to ensure that all entries in the

same subnet have the same routing values. Except for these links and the LOOPBACK link, all links get default BSD values if not specified.

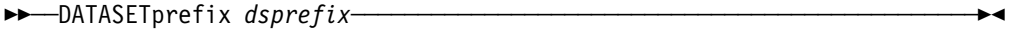
Related Topics

- "GATEWAY Statement" on page 65
- "VARY Command—TCPIP Address Space" on page 96
- Chapter 10, "Configuring the OE RouteD Server" on page 219
- "HOME Statement" on page 71
- "DEVICE and LINK Statement—Virtual Devices (VIPA)" on page 64

DATASETPREFIX Statement

Use the DATASETPREFIX statement to set the high-level qualifier for the dynamic allocation of data sets in TCP/IP.

Syntax



Parameters

dsprefix
The prefix to use as the High-Level Qualifier for the dynamic allocation of data sets. The DATASETPREFIX parameter can be up to 26 characters long and the parameter must NOT end with a period.

Examples

This example shows a DATASETPREFIX statement that sets the high-level qualifier to be TCPIP.V3R3:

```
DATASETPREFIX TCPIP.V3R3
```

Usage Notes

- You can specify DATASETPREFIX in the *hlq*.PROFILE.TCPIP and *hlq*.TCPIP.DATA data sets. The default high-level qualifier distributed with the system is TCPIP.
- DATASETPREFIX in the *hlq*.PROFILE.TCPIP data set is used by the TCPIP address space to qualify dynamically allocated data sets.
- If this statement is used in a profile or configuration data set that is allocated to a client or a server, then that client or server dynamically allocates additional required data sets using the value specified for DATASETPREFIX as the data set name prefix.

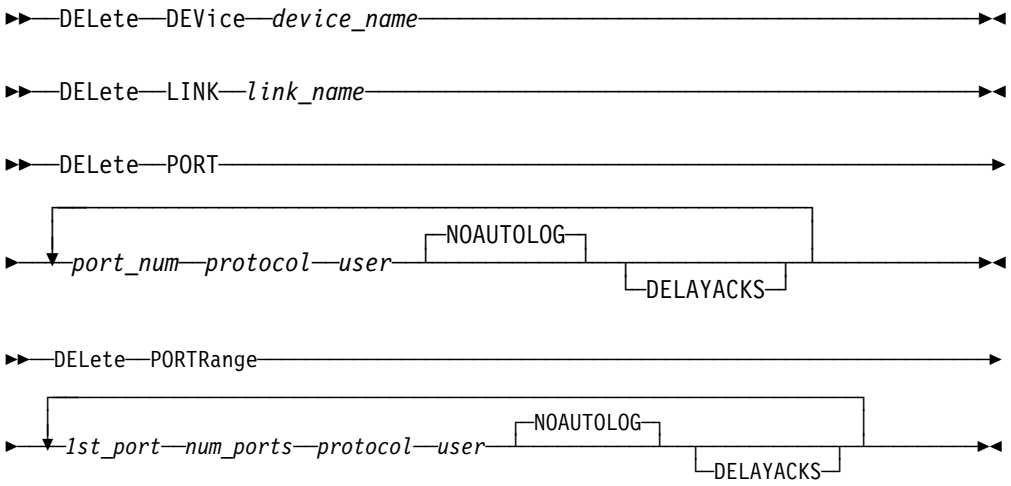
Related Topics

"Configuration Data Sets and HFS Files" on page 5

DELETE Statement

Use the DELETE statement to delete a previously defined device, link, port, or portrange.

Syntax



Parameters

device_name
The name of the device to be deleted. This is the name that was used on a DEVICE statement to define the device to TCP/IP.

link_name
The name of the link to be deleted. This is the name that was used on a LINK statement to define the link to TCP/IP.

port_num
The port number of the port to be deleted. This is the port number that was used on a PORT statement to define the port to TCP/IP.

protocol
Specifies the protocol to be used, either TCP or UDP.

user
The client name associated with the port to be deleted.

NOAUTOLOG
Tells the TCPIP address space ***not*** to restart the server if it was stopped previously.

DELAYACKS
Allows you to alter the default TCP/IP behavior for acknowledgements and delay their transmission so that they can be combined with data sent to the foreign host. This affects acknowledgements returned when a packet is

received with the PUSH bit on in the TCP header. The default behavior is to return an acknowledgement immediately.

The DELAYACKS parameter on the PORT or PORTRANGE statement only applies to the TCP protocol and only affects acknowledgements on this port connection.

1st_port
The first port number of the port range to be deleted. This is the same starting port number used on a PORTRANGE statement to define the port range to TCP/IP.

num_ports
The number of ports to be deleted starting from the *1st_port*. This is the same number of ports that were reserved when the port range was defined with the PORTRANGE statement.

Examples

This example shows DELETE statements that delete a link called sanjose and a device called ourctc:

```
DELETE LINK sanjose
DELETE DEVICE ourctc
```

This example shows a PORT statement to reserve port 5001 for MEGA, and then a DELETE PORT statement to delete the port:

```
PORT 5001 TCP MEGA
DELETE PORT 5001 TCP MEGA
```

This example shows several PORTRANGE statements to reserve ports for MEGA, and then several DELETE PORTRANGE statements to delete the ports:

```
PORTRANGE 5000 10 UDP MEGA
          5100 10 TCP MEGA NOAUTOLOG
          5200 10 UDP MEGA DELAYACKS
          5300 10 TCP MEGA
          5400 10 UDP MEGA
          5500 10 TCP MEGA NOAUTOLOG DELAYACKS
DELETE PORTRANGE
          5000 10 UDP MEGA
          5100 10 TCP MEGA NOAUTOLOG
          5200 10 UDP MEGA DELAYACKS
          5300 10 TCP MEGA
          5400 10 UDP MEGA
          5500 10 TCP MEGA NOAUTOLOG DELAYACKS
```

Usage Notes

- To delete a link, you must first delete any associated HOME entry by specifying a HOME statement that does not include the link.
- To delete a device, you must first delete all associated links.
- To delete a device that has been started, you must first stop the device.
- Individual ports cannot be removed from a PORTRANGE; you must delete the entire PORTRANGE.

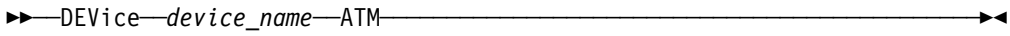
- To delete a specific PORT or PORTRANGE, any parameters specified on the corresponding PORT or PORTRANGE statement must also be specified on the DELETE PORT or DELETE PORTRANGE statement.

DEVICE and LINK Statement—ATM Devices

Use the DEVICE statement to specify the name of the asynchronous transfer mode (ATM) device that you use. Use the LINK statement to define a network interface link associated with the ATM device.

OS/390 TCP/IP OpenEdition does not support the use of an ATM port as a transport facility. ATM DEVICE and LINK statements can only be used to allow SNMP retrieval of network management data for VTAM traffic that is sent over ATM. Therefore, START and STOP of an ATM device are also not supported. For an example of specifying ATM DEVICE and LINK statements for use by SNMP, see "Step 4: Configure the ATM Open Systems Adapter 2 (ATM OSA-2) Support" on page 214.

Syntax

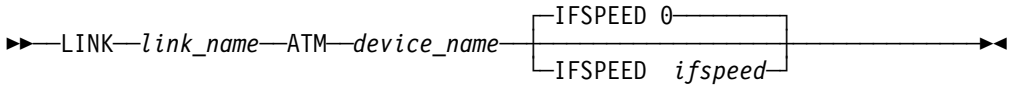


Parameters

device_name
The name of the device. The device name must be the OSA name known to MPC and OSA/SF. The maximum length is 8 characters. The same name is specified in the LINK statement.

ATM
Specifies the device is for ATM use.

Syntax



Parameters

link_name
The unique assigned link name. The link name is the port name known to MPC and OSA/SF. The maximum length is 16 characters.

ATM
Specifies that the link is an ATM link.

device_name
The *device_name* must be the same as specified in the DEVICE statement.

IFSPEED *ifspeed*
An optional estimate of the interface's current bandwidth in bits per second. This value is accessible to SNMP for management queries, but has no effect on operation of the device.

Examples

The following example specifies that USAATM is an ATM device:

```
DEVICE USAATM ATM
```

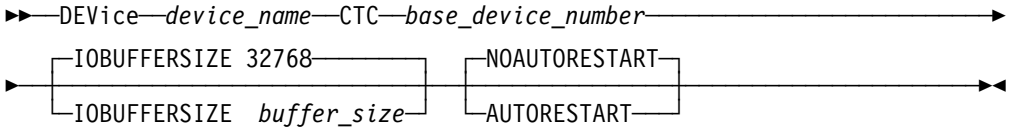
Related Topics

- Chapter 9, “Configuring Simple Network Management Protocol (SNMP) for OE” on page 199
- “BSDROUTINGPARMS Statement” on page 48
- “GATEWAY Statement” on page 65
- “HOME Statement” on page 71
- “START Statement” on page 91
- “VARY Command—TCPIP Address Space” on page 96

DEVICE and LINK Statement—CTC Devices

Use the DEVICE statement to specify the name and hexadecimal device number of the channel-to-channel (CTC) devices that you use. Use the LINK statement to define a network interface link associated with the CTC devices. You must use a separate DEVICE statement for each device you use. The same is true for the LINK statement.

Syntax



Parameters

device_name
The name of the device. The maximum length is 16 characters. The same name is specified in the LINK statement.

CTC
Specifies the device is a channel-to-channel (CTC) device.

base_device_number
The hexadecimal base device number associated with the CTC adapter. Two numbers are used by TCPIP: the *base_device_number* and *base_device_number+1*.

IOBUFFERSIZE *buffer_size*
Specifies the I/O buffer size. The buffer size must be 32K, 32768, or 65535.

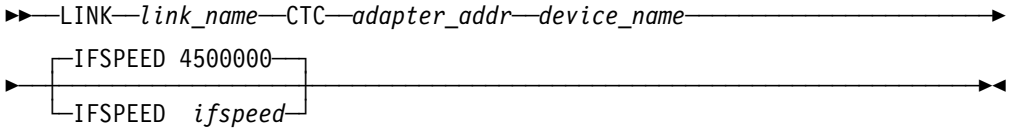
AUTORESTART
In the event of a device failure, the TCP/IP address space will attempt to reactivate the device.

Note:

TCP/IP automatically attempts reactivation of the device following some device-failure indications (regardless of the AUTORESTART setting). Specifying AUTORESTART causes TCP/IP to attempt reactivation following *all* device-failure indications.

NOAUTORESTART
In the event of a device failure, the TCP/IP address space will not attempt to reactivate the device.

Syntax



Parameters

link_name
The unique assigned link name. The maximum length is 16 characters.

CTC
Specifies that the link is a channel-to-channel link.

adapter_addr
An integer used to specify which device number is the read device number and which device number is the write device number. Use 0 to indicate that the base device number is the read device and 1 to indicate that the base device number is the write device.

device_name
The *device_name* must be the same as specified in the DEVICE statement.

IFSPEED *ifspeed*
An optional estimate of the interface's current bandwidth in bits per second. This value is accessible to SNMP for management queries, but has no effect on operation of the device.

Usage Notes

- You can add new DEVICE and LINK statements using the VARY TCPIP command. You can also delete and redefine existing statements.

  When you add new LINK statements, any corresponding GATEWAY, HOME, and TRANSLATE statements coded to include the new links are treated as replacements for active statements. Therefore, when you code the GATEWAY, HOME, or TRANSLATE statements in the OBEYFILE data set, be sure to include existing links that you want to have active in your configuration.

- The configured I/O buffer sizes at each end of the CTC connection must match. A buffer size mismatch can cause packet loss or I/O errors, resulting in deactivation of the CTC connection.

  With OS/390 TCP/IP OpenEdition, CTC I/O buffer size may be explicitly specified with the IOBUFFERSIZE parameter. In previous releases, the CTC I/O buffer size was the larger of 32768 or Lrg_Env_Size specified on the LARGEENVELOPEPOOLSIZE statement. The LARGEENVELOPEPOOLSIZE statement is now obsolete.

Related Topics

- “BSDROUTINGPARMS Statement” on page 48
- “GATEWAY Statement” on page 65
- “HOME Statement” on page 71
- “START Statement” on page 91
- “VARY Command—TCPIP Address Space” on page 96

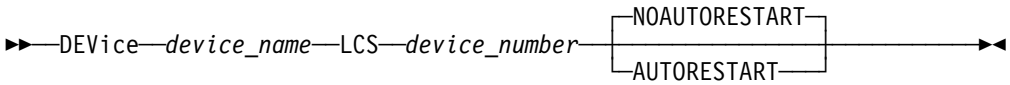
DEVICE and LINK Statement—LAN Channel Station Devices

Use the DEVICE statement to specify the name and hexadecimal device number of an IBM 8232 LAN channel station (LCS) device or an IBM 3172 Interconnect Controller.

Use the LINK statement to define a network interface link associated with an LCS device. The LINK statements used are the Ethernet Network LCS LINK statement, the Token-Ring Network or PC Network LCS LINK statement,and the FDDI LCS LINK statement.

You must use a separate LINK statement for each link associated with an LCS device.

Syntax



Parameters

device_name
The name of the device. The maximum length is 16 characters. The same name is specified on the LINK statements.

LCS
Specifies the device is a LAN Channel Station.

device_number
The hexadecimal device number of the LCS. *device_number*+1 is also used by the TCPIP address space.

AUTORESTART
In the event of a device failure, the TCP/IP address space will attempt to reactivate the device.

Note:

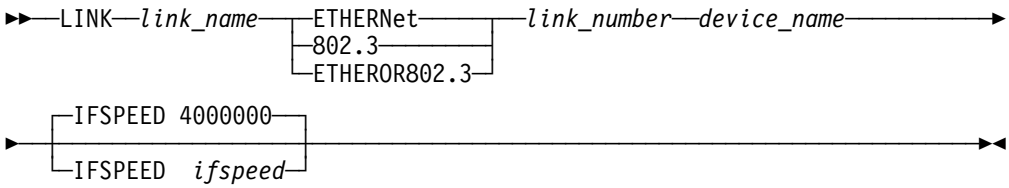
TCP/IP automatically attempts reactivation of the device following some device-failure indications (regardless of the AUTORESTART setting). Specifying AUTORESTART causes TCP/IP to attempt reactivation following *all* device-failure indications.

NOAUTORESTART
In the event of a device failure, the TCP/IP address space will not attempt to reactivate the device.

LINK Statement for Ethernet Network LCS

This LINK statement is used to define an Ethernet link on an IBM 3172 Interconnect Controller and IBM 8232 LAN Channel Station (LCS) device.

Syntax



Parameters

link_name
The unique assigned link name. The maximum length is 16 characters.

link_number
Is an integer: 0 for the first Ethernet protocol network in the LCS, 1 for the second Ethernet protocol network, and so on.

device_name
The *device_name* must be the same name as specified in the DEVICE statement.

IFSPEED *ifspeed*
An optional estimate of the interface's current bandwidth in bits per second. This value is accessible to SNMP for management queries, but has no effect on operation of the device.

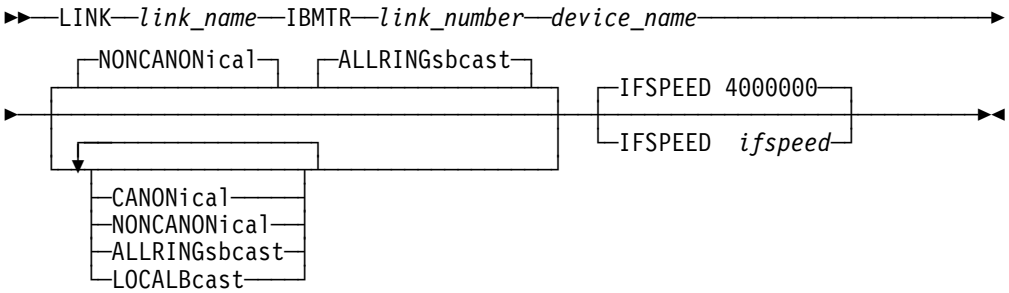
LINK Statement for Token-Ring Network or PC Network LCS

The token-ring LCS LINK statement is used to define the token-ring link to the LCS (IBM 8232 or IBM 3172) previously defined by the LCS DEVICE statement. By default, the token-ring LCS LINK statement is also used to define the PC Network link.

Medium Access Control (MAC) addresses in the Address Resolution Protocol (ARP) packets on this token-ring network are in the more common, non-canonical format.

Note: All TCPIP hosts and gateways on a given token-ring network must be configured to use the same form for MAC addresses in ARP packets, either canonical or non-canonical. For more information about the terms, canonical and non-canonical, see IEEE standards 802.3 and 802.5.

Syntax



Parameters

link_name
The unique assigned link name. The maximum length is 16 characters.

IBMTR
Specifies that the link is to an IBM Token Ring.

link_number
Is 0 for the first token ring in the LCS, 1 for the second token ring, and so on.

device_name
The *device_name* must be the same as specified in the DEVICE statement.

CANONICAL
MAC addresses in Address Resolution Protocol (ARP) packets on this token-ring network are in the canonical IEEE 802.5 form.

NONCANONICAL
MAC addresses in ARP packets on this token-ring network are in the more common non-canonical format. This is the default.

ALLRINGSBCAST
All IP and ARP broadcasts are sent as all-rings broadcasts, which are propagated through token-ring bridges. This is the default.

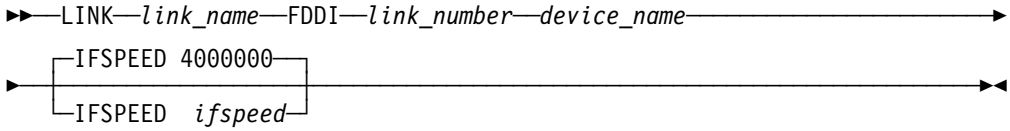
LOCALBCAST
All IP and ARP broadcasts are sent only on the local ring and are not propagated through token-ring bridges.

IFSPEED *ifspeed*
An optional estimate of the interface's current bandwidth in bits per second. This value is accessible to SNMP for management queries, but has no effect on operation of the device.

LINK Statement for FDDI LCS

This LINK statement is used to define the Fiber Distributed Data Interface (FDDI) link to the LCS (IBM 3172 Models 002 and 003) defined by the LCS DEVICE statement.

Syntax



Parameters

link_name
The unique assigned link name. The maximum length is 16 characters.

FDDI
Specifies that the link is to an FDDI network.

link_number
Is 0 for the first FDDI adapter in the LCS, 1 for the second FDDI adapter, and so on.

device_name
The *device_name* must be the same as specified in the DEVICE statement.

IFSPEED *ifspeed*
An optional estimate of the interface's current bandwidth in bits per second. This value is accessible to SNMP for management queries, but has no effect on operation of the device.

Examples

- In this example, LCS1 is a 3172 model 1 with a Token Ring and Ethernet adapter.

```
DEVICE LCS1   LCS           BA0
LINK TR1   IBMTR    0 LCS1
LINK ETH1  ETHERNET 1 LCS1
```

- In this example, LCS2 is a 3172 model 2 with a FDDI adapter.

```
DEVICE LCS2   LCS           BE0
LINK FDDI1 FDDI     0 LCS2
```

- This example shows how you might code DEVICE, LINK, and related statements for an LCS connection.

```
DEVICE LCS1 LCS BA0
LINK TR1  IBMTR    0 LCS1
LINK TR2  IBMTR    1 LCS1 LOCALBCAST
LINK ETH1 ETHERNET 0 LCS1
HOME
   192.10.10.10  TR1
   9.67.43.10    TR2
   128.50.17.1   ETH1

   GATEWAY
;
; Network  First hop   Driver   Packet size  Subnet mask  Subnet value

  192.10.10    =         TR1       2000       0
  9            =         TR2       2000       0.255.255.0  0.67.43.0
  128.50       =         ETH1      1500       0.0.240.0    0.0.16.0
  DEFAULTNET 9.67.43.1   TR2    DEFAULTSIZE   0

; The following BSDROUTINGPARMS statement would be used if running ORouteD.
; If not running ORouteD, use prior gateway stats.
;
; ;  link     maxmtu   metric   subnet mask     dest addr
; BSDROUTINGPARMS false
;    TR1       2000       0     255.255.255.0   0
;    TR2       2000       0     255.255.255.0   0
;    ETH1      1500       0     255.255.240.0   0
; ENDBSDROUTINGPARMS
;

START LCS1
```

Usage Notes

- You can add new DEVICE and LINK statements using the VARY TCPIP command. You can also delete and redefine existing statements.

  When you add new LINK statements, any corresponding GATEWAY, HOME, and TRANSLATE statements coded to include the new links are treated as replacements for active statements. Therefore, when you code the GATEWAY, HOME, or TRANSLATE statements in the OBEYFILE data set, be sure to include existing links that you want to have active in your configuration.

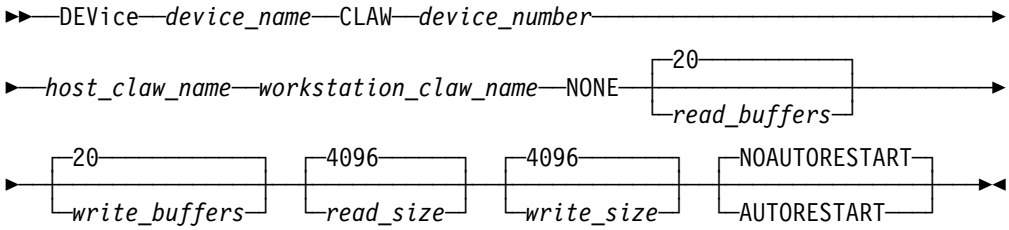
Related Topics

- “BSDROUTINGPARMS Statement” on page 48
- “GATEWAY Statement” on page 65
- “HOME Statement” on page 71
- “START Statement” on page 91
- “VARY Command—TCPIP Address Space” on page 96

DEVICE and LINK Statement—CLAW Devices

Use the DEVICE statement to specify the name and hexadecimal device number of a CLAW device that you use. Devices that use the CLAW protocol include RISC System/6000 and SP2. Only one DEVICE statement should be used for each device. Use the LINK statement to define a network interface link associated with CLAW devices. Only one LINK statement should be used for each device.

Syntax



Parameters

device_name
The name of the device. The maximum length is 16 characters. The same name is specified in the LINK statements.

CLAW
Specifies the device is a CLAW device.

device number
The hexadecimal device number of the RISC System/6000. TCPIP also uses device number + 1.

host_claw_name
A value that defines the name of the host system in the system validation exchange between the TCPIP code and the workstation code. This name must be the same name that is defined as the CLAW mode HOST name in SMIT on the RISC/6000. The maximum length is 8 characters.

workstation_claw_name
A value for the name of the workstation for the system validation exchange. This value must be the name of the CLAW mode adapter defined in SMIT on the RISC/6000 .The maximum length is 8 characters.

NONE
A place holder reserved for future use.

read_buffers
This is the decimal number (one or more) of buffers to allocate to the read channel program. This should be large enough to give TCPIP sufficient time to process the received data and append the buffer to the running channel program before it terminates. Each of these buffers uses real storage, so the number should be small enough not to impact overall system performance. The default is 20.

write_buffers
This is the decimal number (one or more) of buffers to allocate to the write channel program. This should be large enough that a busy TCPIP can reuse buffers without the channel program terminating. Each of these buffers uses real storage, so the number should be small enough not to impact overall system performance. The default is 20.

read_size
This is the size of the read buffers. Values are:

1024
2048
3072
4096

The value must be greater than or equal to the transmit buffer size specified in the RISC System/6000. The default is 4096. The value must be 4096 for ESCON or RISC System/6000.

write_size
This is the size of the write buffers. Values are:

1024
2048
3072
4096

The value must be less than or equal to the receive buffer size specified in the RISC System/6000. The default is 4096. The value must be 4096 for ESCON or RISC System/6000.

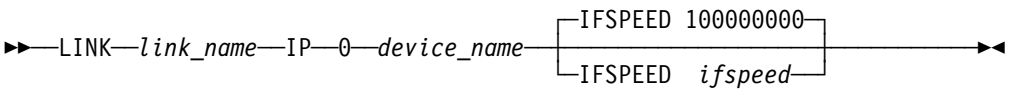
AUTORESTART
In the event of a device failure, the TCP/IP address space will attempt to reactivate the device.

Note:

TCPIP automatically attempts reactivation of the device following some device-failure indications (regardless of the AUTORESTART setting). Specifying AUTORESTART causes TCPIP to attempt reactivation following *all* device-failure indications.

NOAUTORESTART
In the event of a device failure, the TCP/IP address space will not attempt to reactivate the device.

Syntax



Parameters

link_name
The unique assigned link name. The maximum length is 16 characters.

IP A constant.

0 A constant.

device_name
The *device_name* must be the same as specified in the DEVICE statement.

IFSPEED *ifspeed*
An optional estimate of the interface's current bandwidth in bits per second. This value is accessible to SNMP for management queries, but has no effect on operation of the device.

Examples

This example shows how you might code DEVICE, LINK, and related statements for a RISC System/6000 connection.

```
DEVICE RS6K CLAW 6B2 HOST PSCA NONE
LINK IPLINK1 IP 0 RS6K
HOME
   192.10.10.1 IPLINK1

GATEWAY
;
; Network  First hop   Driver   Packet size  Subnet mask  Subnet value

  192.10.10.2  =         IPLINK1 DEFAULTSIZE  HOST
  DEFAULTNET 192.10.10.2 IPLINK1 DEFAULTSIZE  0

; The following BSDROUTINGPARMS statement would be used if running ORouteD
;
;   link     maxmtu   metric   subnet mask     dest addr
 BSDROUTINGPARMS false
    IPLINK1   2000       0     255.255.255.0   192.10.10.2
 ENDBSDROUTINGPARMS;

START RS6K
```

Usage Notes

- You can add new DEVICE and LINK statements using the VARY TCPIP command. You can also delete and redefine existing statements.

  When you add new LINK statements, any corresponding GATEWAY, HOME, and TRANSLATE statements coded to include the new links are treated as replacements for active statements. Therefore, when you code the GATEWAY, HOME, or TRANSLATE statements in the OBEYFILE data set, be sure to include existing links that you want to have active in your configuration.

- If the OS/390 server running the CLAW device driver is a second-level host on a first-level VM system, the OS/390 server should be defined as V=R (virtual=real). Unpredictable results might occur if CLAW is run without V=R, as dynamic changes to the channel program (performed by the CLAW driver) will not be observed by the channel.

Related Topics

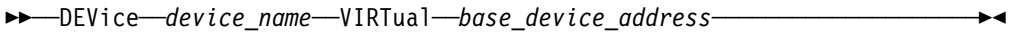
- “BSDROUTINGPARMS Statement” on page 48
- “GATEWAY Statement” on page 65
- “HOME Statement” on page 71
- “START Statement” on page 91
- “VARY Command—TCPIP Address Space” on page 96

DEVICE and LINK Statement—Virtual Devices (VIPA)

Use the DEVICE statement to specify a device name and virtual number of a virtual device.

Use the LINK statement to define the link on the DEVICE statement.

Syntax



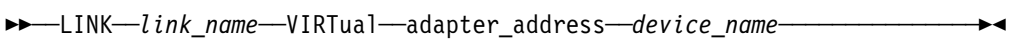
Parameters

device_name
The name of the device. The maximum length is 16 characters. The same name is specified in the LINK statement.

VIRTUAL
Specifies that the device is not associated with real hardware and is used for fault tolerance support. The virtual devices always stay active and are never subject to physical failure.

base_device_address
The hexadecimal base device associated with the virtual device. Addresses start at zero and are incremented by 1 for each device.

Syntax



Only one LINK statement can be defined for each virtual device.

Parameters

link_name
The unique assigned link name. The maximum length is 16 characters. The same name is specified in the HOME statement.

VIRTUAL
Specifies that the link is a virtual link that is not associated with real hardware and is used for fault tolerance support.

adapter_address
The adapter address is always zero. No more than one virtual link can be specified for a virtual device.

device_name
The *device_name* must be the same as specified in the DEVICE statement.

Examples

```
DEVICE VDEV1  VIRTUAL 0
LINK   VLINK1 VIRTUAL 0 VDEV1
DEVICE VDEV2  VIRTUAL 1
LINK   VLINK2 VIRTUAL 0 VDEV2
```

Usage Notes

- You can add new DEVICE and LINK statements using the VARY TCPIP command. You can also delete and redefine existing statements.

  When you add new LINK statements, any corresponding GATEWAY, HOME, and TRANSLATE statements coded to include the new links are treated as replacements for active statements. Therefore, when you code the GATEWAY, HOME, or TRANSLATE statements in the OBEYFILE data set, be sure to include existing links that you want to have active in your configuration.
- Only one virtual link can be defined for a virtual device.
- More than one virtual DEVICE/LINK statement can be defined to allow for multiple virtual IP addresses on one TCP/IP image in one MVS system.
- A virtual LINK cannot be coded on the START, GATEWAY or TRANSLATE statements, but can be coded on a BSDROUTINGPARMS statement for interface characteristics such as subnet mask.
- For rules on defining virtual IP addresses for virtual links, see the HOME statement on page “HOME Statement” on page 71.

Related Topics

- “BSDROUTINGPARMS Statement” on page 48
- “HOME Statement” on page 71

GATEWAY Statement

Use the GATEWAY statement to add static routes to the IP route table.

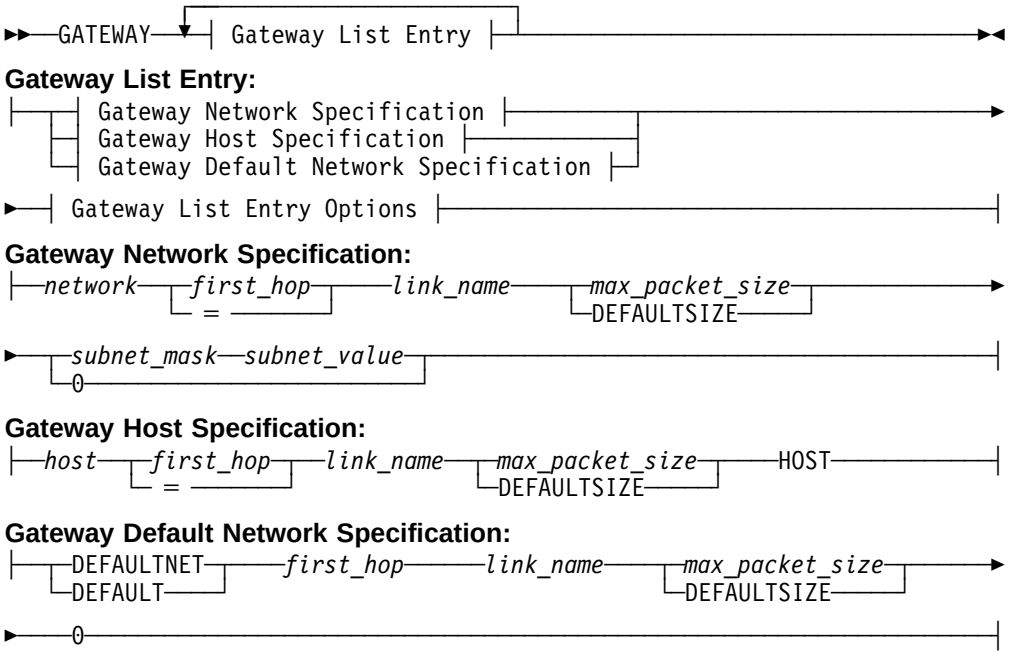
The IP route table can be modified by replacing the table using the VARY TCPIP command, adding new routes using the ORouteD server, or by incoming ICMP redirect packets sent from adjacent machines.

The first GATEWAY statement of each configuration data set executed replaces the existing routing table with the new gateway information. All static routes are deleted, but routes created by ORouteD are not deleted. Subsequent GATEWAY statements in the same data set add entries to the routing table.

Notes:

1. If ORouteD is running, static routes defined by the GATEWAY statement cannot be deleted by ORouteD. If you want ORouteD to manage all routes, an empty GATEWAY statement can be used to eliminate the static routes. ORouteD will find out about them dynamically. See Chapter 10, “Configuring the OE RouteD Server” on page 219 for further explanation of using the GATEWAY statement with ORouteD.
2. Virtual and ATM links are not allowed on the GATEWAY statement.
3. When an incorrect GATEWAY statement entry is encountered, if the remaining entries on that GATEWAY statement can be syntactically checked, they are processed. Otherwise, they are ignored. Subsequent GATEWAY statements in the same profile or obeyfile are processed.
4. A specific host route takes precedence over a network route.

Syntax



Parameters

network
The IP address in dotted-decimal form.

- An example of a class A network is 9.0.0.0.
- An example of a class B network is 129.34.0.0.
- An example of a class C network is 192.9.100.0.

Use the *subnet_mask* and *subnet_value* fields to define a subnetted network.

DEFAULTNET
Specifies the default to use for any network not explicitly routed. DEFAULTNET can be specified only once.

For the DEFAULTNET keyword to take effect, you must specify a GATEWAY statement that defines the route to the *first_hop*.

DEFAULT
Multiple DEFAULT entries can be specified, allowing for multiple default routes in addition to the DEFAULTNET route. If the DEFAULTNET route does not exist or is not active (that is, the link it is defined to is not active), the first active DEFAULT route is used when no specific route matches the destination or source IP address.

host
The host address, specified as 4 octets (192.9.100.3, for example). If a host address is specified, the keyword HOST must be specified in place of the *subnet_mask* field, and the *subnet_value* field must not be specified.

first_hop
Specify one of the following:

- An equal sign (=), meaning that messages are routed directly to destinations on that network or directly to that host. This is not supported for DEFAULTNET or DEFAULT.
- The internet address of a gateway or router that you can reach directly, and that forwards messages for the destination network or host.

link_name
The name of the link through which packets are sent to the specified network. The link name is defined in a LINK statement. It cannot be a VIPA or ATM link type.

max_packet_size
The maximum transmission unit (MTU) in bytes for the network or host. This value can be up to 65535.

The special entry DEFAULTSIZE in this field requests that TCPIP supply a default value of 576. We recommend you use the following sizes instead of DEFAULTSIZE as the packet size for these networks:

1492 bytes for Ethernet 802.3
1500 bytes for Ethernet Version 2 IEEE
1500, or 2000 or 4000 bytes for token ring
4352, or 2000 or 4000 bytes for FDDI
65527 bytes for CTC
4096 bytes for CLAW

You can change these values as needed later during performance tuning as explained in TCP/IP: Performance Tuning Guide

subnet_mask
A bit mask (expressed in dotted-decimal form) that shows the bits of the host field that make up the subnet. The usual practice is to make the bits contiguous in the host field. The network field must be zero.

If the network does not use subnets, specify a *subnet_mask* of 0 and omit the *subnet_value*.

If this is a host entry, specify a *subnet_mask* of HOST and omit the *subnet_value*.

subnet_value
Value of each *subnet_mask*. Each subnet should have a unique dotted-decimal representation. A value of 0 indicates the default route for any subnet of this

network that is not specifically routed. Do not include the *subnet_value* field if the *subnet_mask* field is 0 or HOST.

If the network has one or more subnets, specify a separate entry in the GATEWAY statement for each subnet. The network part of each GATEWAY entry is identical (contains the IP network address as if the network has no subnets). The *subnet_mask* part of each GATEWAY entry is also identical, but the *subnet_value* varies.

Examples

1. This is an example of a network and the corresponding GATEWAY statements.

   Figure 1 shows a host, `MVS1`, directly connected to networks 193.9.200 and 193.0.2. Neither network has subnets. `MVS1` is indirectly connected to network 128.84, which has subnets using the high-order byte of the host number as the subnet field. The subnet 128.84.1 is accessible through 193.9.200.2; the subnet 128.84.55 is accessible through 193.9.200.100; and the other subnets of 128.84 are accessible through 193.0.2.2. All packets destined for a network that has no entry in the routing table should be routed to 193.0.2.3. All packets to the host `jakespc` should be routed through 193.0.2.2.

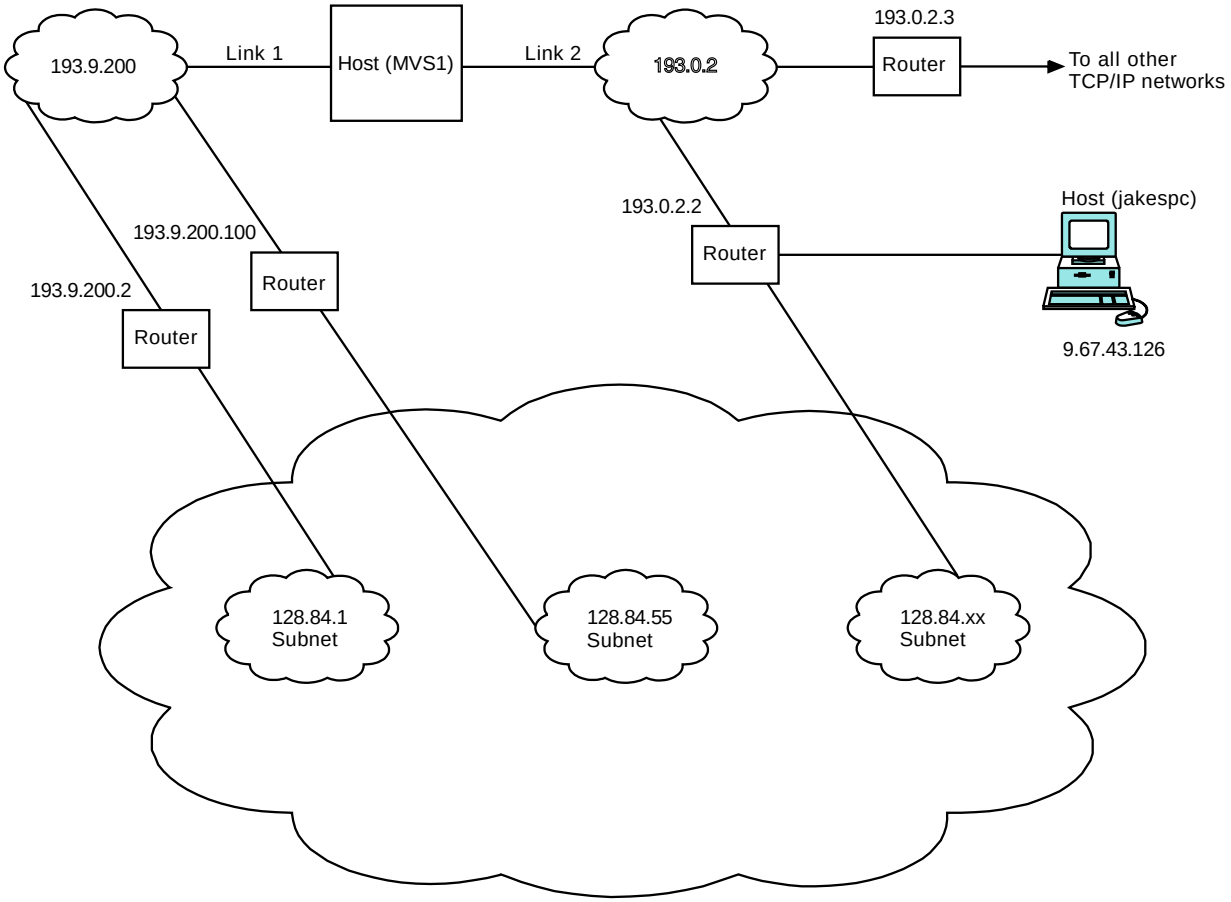


Figure 1. Example of Network Connectivity

The following is an example of the corresponding GATEWAY statement.

```
GATEWAY
*net_number first_hop link_name packet_size subnet_mask  subnet_value
 193.9.200          =        LINK1  DEFAULTSIZE  0
 193.0.2            =        LINK2  DEFAULTSIZE  0
 128.84        193.9.200.2   LINK1  DEFAULTSIZE  0.0.255.0   0.0.1.0
 128.84        193.9.200.100 LINK1  DEFAULTSIZE  0.0.255.0   0.0.55.0
 128.84        193.0.2.2     LINK2  DEFAULTSIZE  0.0.255.0   0
 DEFAULTNET    193.0.2.3     LINK2  DEFAULTSIZE  0
 9.67.43.126   193.0.2.2     LINK2  DEFAULTSIZE  HOST
```

Note:

For the networks not using subnets, the *subnet_mask* is 0 the *subnet_value* is omitted.

The *subnet_mask* is the same for all instances of the subnet network 128.84. A subnet value of 0 in the *subnet_mask* field indicates the default route for the subnets not explicitly routed.

2. This example shows a GATEWAY statement that is divided by the types of routes used.

```
GATEWAY
;
; Direct Routes - Routes that are directly connected to my interfaces.
;
; Network  First hop  Link name Packet size  Subnet mask  Subnet value
  193.9.200          =        LINK1  DEFAULTSIZE  0
  193.0.2            =        LINK2  DEFAULTSIZE  0;
; Indirect Routes - Routes that are reachable through routers on my
;                   network.
;
; Network  First hop  Link name Packet size  Subnet mask  Subnet value

  128.84        193.9.200.2   LINK1  DEFAULTSIZE  0.0.255.0   0.0.1.0
  128.84        193.9.200.100 LINK1  DEFAULTSIZE  0.0.255.0   0.0.55.0
  128.84        193.0.2.2     LINK2  DEFAULTSIZE  0.0.255.0   0
  9.67.43.126   193.0.2.2     LINK2  DEFAULTSIZE  HOST
;
;
; Default Route - All packets to an unknown destination are routed
;                 through this route.
;
; Network  First hop  Link name Packet size  Subnet mask  Subnet value

  DEFAULTNET    193.0.2.3     LINK2  DEFAULTSIZE  0
```

Usage Notes

- Packet size considerations:
  - A network IP address is distinguished from a host IP address by determining if the host portion of the IP address is all zeroes as determined by the IP address class:
    - Class A — *a*.0.0.0 where 0<=*a*<=127
    - Class B — *a.b*.0.0 where 128<=*a*<=191, 0<=*b*<=255
    - Class C — *a.b.c*.0 where 192<=*a*<=223, 0<=*b*<=255, 0<=*c*<=255

Therefore, it is not valid to have a host (or source or destination IP address) defined at an internet address containing a zero for the host (for example, 127.0.0.0).

– Information is transferred over a TCP connection in discrete packets. Each packet includes a TCP header and an IP header. The header size is independent of the amount of user information included, so the larger the packets sent, the less relative bandwidth is consumed by protocol headers. Also, the TCP software consumes a fixed amount of CPU time for each packet, independent of the packet size.

– The *max_packet_size* that OS/390 TCP/IP OpenEdition can handle varies for different networks. For example, while the largest packet size for the Ethernet protocol is 1500 bytes, the largest packet size for the 802.3 protocol is 1492 bytes.

– The actual packet size will be determined by the total network connection.

  - If a locally-attached host has a packet size smaller than yours, transfers to that host will use the smaller size.

  - The TCP maximum segment size for the 3172 Interconnect Controller Program is 4096. Any packet specifications over 4096 are limited by this restriction. For example, if you specified a packet size of 4352, the resulting packet size would still only be 4096 + the header = 4132.

– Large packets can be fragmented by intervening gateways. Fragmentation and reassembly of packets are expensive in their use of bandwidth and CPU time. Therefore, packets sent through gateways to other networks should use the default size, DEFAULTSIZE, unless all intervening gateways and networks are known to accept larger packets.

– If this is a RISC System/6000 link, then the *max_packet_size* cannot exceed the *write_size* specified on the corresponding DEVICE statement.

– Occasionally, your packets will pass through routers that fragment packets to the internet default size (576 bytes). Use the GATEWAY configuration statement to further reduce packet sizes. For example, the router to network 192.8.4 is reached through router 14.0.0.10., and somewhere along the path, packets larger than 460 bytes are fragmented. Throughput can be improved using the following GATEWAY statement:

```
GATEWAY
; Network   first-hop   link   packet-size    subnet-mask
  192.8.4   14.0.0.10   LINK1     460            0
```

- If the gateway table is empty, the LOOPBACK test address is still routed properly. For information on testing commands with LOOPBACK, see the *OS/390 TCP/IP OpenEdition User's Guide*.

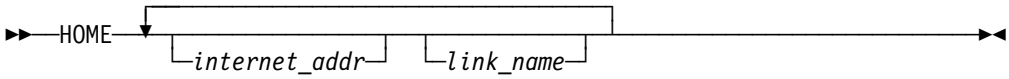
Related Topics

- "DEVICE and LINK Statements" on page 36
- "VARY Command—TCPIP Address Space" on page 96
- Chapter 10, "Configuring the OE RouteD Server" on page 219
- "BSDROUTINGPARMS Statement" on page 48

HOME Statement

The HOME statement provides the list of home addresses and associated link names (called the HOME list).

Syntax



Parameters

internet_addr
The internet address valid for this host. The internet address can be associated with a real or virtual link. The internet address must be specified in dotted-decimal form.

link_name
The name of the link defined in a previous LINK statement that is associated with the home address.

Examples

This example shows a HOME statement that defines the IP addresses of each link in the host. The corresponding GATEWAY statement is shown in Example 2 on page 69.

```
HOME
   192.1.1.1      VLINK1
   130.50.75.1    TR1
   193.5.2.1      ETH1
   192.2.1.1      VLINK2
   9.67.43.110    FDDI1
```

VLINK1 and VLINK2 are examples of virtual links associated with virtual IP addresses, while others are examples of real links associated with real IP addresses.

The following example shows the definition of an additional LOOPBACK address:

```
HOME  9.67.113.105   CTCD00   ; CTC IP address for this system
      14.0.0.0       LOOPBACK ; additional LOOPBACK address
```

Usage Notes

- Only one home address can be associated with a link. If the same link is specified in more than one HOME list entry, only the home address in the last entry is associated with the link. The only exception to this is the LOOPBACK link. Multiple home entries are accepted for LOOPBACK in addition to 127.0.0.1.

  A *link_name* of LOOPBACK defines the IP address to use for LOOPBACK. No DEVICE or LINK statement is needed for LOOPBACK, and it cannot be started or stopped (LOOPBACK is always running and in a started state).

You can use LOOPBACK in the HOME list only to define additional LOOPBACK addresses. If you try to redefine the default LOOPBACK address of 127.0.0.1, it is flagged as a duplicate entry.

- If the PRIMARYINTERFACE statement is not specified, then the first address in the HOME list is the default local address. A default local address is used for the GETHOST() function.
- The first HOME statement of each configuration data set executed replaces the existing HOME list with the new list; subsequent HOME statements in the same data set add entries to the list.
- If the first HOME statement of a profile contains no internet address or link name, all addresses are removed from the HOME list except for the loopback address.
- When an incorrect HOME entry is encountered, all entries following that entry on that HOME statement are ignored. Subsequent HOME statements are processed.
- If using ORouteD, define no more than 255 home entries.
- When defining virtual IP addresses, observe the following rules and recommendations:

  1. You should specify the primary virtual IP address first in the HOME list or specify it on the PRIMARYINTERFACE statement. The remainder of the virtual links can be specified in any order.
  2. If subnetting is not used (that is, there is no subnet mask), the network portion of any virtual IP address must not be the network portion of any real link addresses in the network.
  3. If subnetting is used, the subnetwork portion of any virtual IP address must not be the subnetwork portion of any real link addresses in the network. Furthermore, if the network portion of any virtual IP address is different from the network portion of any real link addresses in the OS/390 server, the network restriction rule described in number 2 must apply.
  4. The network or subnetwork portions of virtual IP addresses can be the same across multiple TCP/IP images in the network. If they are the same, you must enable Host Route Broadcasting using the ORouteD -h or -hv parameters. Also, adjacent routers must support Host Route Broadcasting.
  5. If the real link addresses are subnetted in one network, you should assign a new subnetwork for the virtual link. If subnetwork addresses are not available, then you should use a new network address for the virtual link, preferably a class C address.
  6. More than one virtual IP address can be defined in one network or subnetwork.
  7. The virtual IP address can be used as the primary or only destination for the name of an OS/390 server on the domain name server. A workstation on the network would use the OS/390 server name (translated into the virtual IP address) to access applications on the OS/390 server.
  8. To obtain non-disruptive TCP-connection fault tolerance, use ORouteD which provides dynamic routing based on the Routing Information Protocol (RIP). See Chapter 10, “Configuring the OE RouteD Server” on page 219 for more information.
  9. For ATM devices, the HOME statement can be used to set an IP address in an ATM port through SNMP. For an example of this, see “Step 4: Con-

figure the ATM Open Systems Adapter 2 (ATM OSA-2) Support" on page 214.

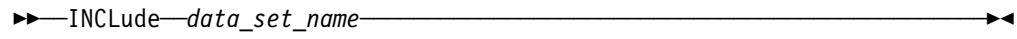
Related Topics

- "DEVICE and LINK Statements" on page 36
- "PRIMARYINTERFACE Statement" on page 87
- "BSDROUTINGPARMS Statement" on page 48

INCLUDE Statement

This statement causes another data set that contains profile configuration statements to be included and processed at the point it occurs in the including file stream.

Syntax



Parameters

data_set_name
A fully qualified data set name that identifies a sequential file. The sequential file can be a sequential data set or a PDS with the member name. It can not be an HFS file.

Examples

The example below shows a profile that includes two other profiles:

```
; Include device/link/home/gateway/start for appropriate device
INCLUDE USER.INCLUDE(CTC)
;
; ARP age of 20 minutes
ARPAGE 20
;
; Default assorted parms
ASSORTEDPARMS
   IGNOREREDIRECT
   NOFWD
   NOUDPQUEUELIMIT
ENDASSORTEDPARMS
;
; Default IP Config parms
IPCONF ARPTO 1200 CLAWUSED STOPON NODATAGR IGNORER REASSEMBL 60 TTL 64
;
; Default Interval
KEEPALIVEOPTIONS
   INTERVAL 120
ENDKEEPALIVEOPTIONS
;
; Include sockets maximum connectors
INCLUDE USER.INCLUDE.MAXCON
;
; Default TCP Config parms
TCPCONF INT 120 UNRESTRICTL TCPSENDB 65535 TCPRCVB 65535
;
; Default UDP Config parms
UDPCONF UNRESTRICTL UDPCHK UDPSENDB 65535 UDPRCVB 65535 NOUDPQ
```

Below is the INCLUDE file, USER.INCLUDE(CTC), that defines the CTC device:

```
; CTC Device
DEVICE CTC1     CTC     D00  IOBUFFERSIZE 32K
LINK   CTCD00   CTC  0  CTC1 IFSPEED 4500000
;
HOME  9.67.113.105   CTCD00 ; MVSVIC04 - MVSVIC02
;
; Direct routes
GATEWAY
; Network  First hop  Driver  Packet size  Subnet mask    Subnet value
9.67.116.124   =      CTCD00       576        HOST
;
 START CTC1
```

Below is the INCLUDE file, USER.INCLUDE.MAXCON, that defines maximum socket connections:

```
; Socket connections default is 10, want to run with more
SOMAXCON 50
```

Usage Notes

The TRUNC statement applies only to the file being processed and not to the file being included.

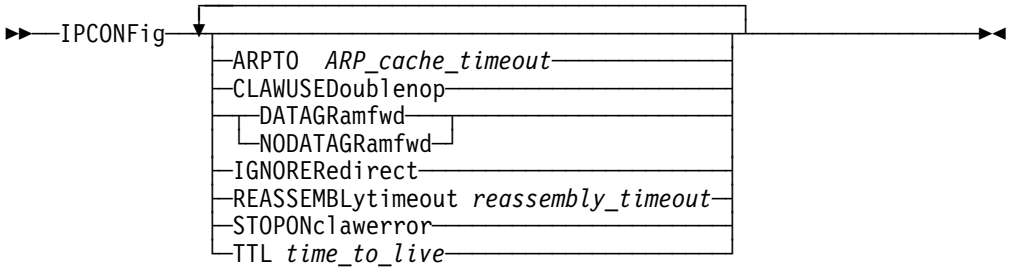
Related Topics

"TRUNC Statement" on page 95

IPCONFIG Statement

Use the IPCONFIG statement to update the IP layer of TCP/IP.

Syntax



Parameters

ARPTO

Use ARPTO to specify the number of seconds between creation or revalidation and deletion of ARP table entries. An ARP table entry is revalidated when another ARP packet is received from the same host specifying the same hardware address.

This parameter serves the same purpose as the ARPAGE statement, but the value specified on ARPAGE is in minutes while the value specified on the ARPTO parameter is in seconds.

CLAWUSEDOUBLENOP

Forces channel programs for CLAW devices to have two NOP CCWs to end the channel programs. This is required for some vendor devices, and applies to only first-level MVS systems. The CLAWUSEDOUBLENOP parameter is confirmed by the message:

```
CLAWUSEDOUBLENOP is set
```

DATAGRAMFWD

Enables the transfer of data between networks. The DATAGRAMFWD parameter is confirmed by the message:

```
IP forwarding is enabled
```

NODATAGRAMFWD

Stops the transfer of data between networks by disabling IP datagram routing between different network interfaces. This statement can be used for security or to ensure correct usage of limited resources. The NODATAGRAMFWD parameter is confirmed by the message:

```
IP forwarding is disabled.
```

If either ASSORTEDPARMS NOFWD or IPCONFIG NODATAGRAMFWD is specified in a profile, or if neither the ASSORTEDPARMS nor the IPCONFIG statement is specified, forwarding is disabled. If the ASSORTEDPARMS or

IPCONFIG statement is specified and the NOFWD and NODATAGRAMFWD parameters are not specified, forwarding is enabled.

IGNOREREDIRECT
Causes TCP/IP to ignore ICMP Redirect packets. The IGNOREREDIRECT parameter is confirmed by the message:

```
ICMP will ignore redirects
```

If you are using ORouteD, use this option since ORouteD does not support ICMP redirects.

REASSEMBLYTIMEOUT *reassembly_timeout*
IP reassembly time-out value in seconds.

STOPONCLAWERROR
Stops channel programs (HALTIO and HALTSIO) when a device error is detected. The STOPONCLAWERROR parameter is confirmed by the message:

```
STOPONCLAWERROR is set
```

TTL *time_to_live*
IP time to live or hop count.

Examples

This example shows an IPCONFIG statement which causes ARP table entries to be deleted 2400 seconds after creation or revalidation, forces channel programs for CTC devices to have two NOP CCWs to end the channel programs, disables IP forwarding, and causes TCP/IP to halt on certain CLAW errors.

```
IPCONFIG ARPTO 2400 CLAWUSED NODATAGR STOPON
```

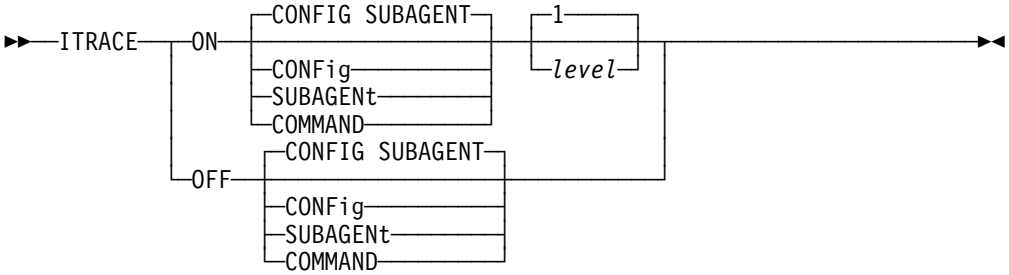
Usage Notes

If any of the IPCONFIG statement parameters are specified on other statements, such as ASSORTEDPARMS, the settings from the last statement processed are used. You will see an appropriate message.

ITRACE Statement

Use the ITRACE statement to control TCP/IP run-time tracing.

Syntax



Parameters

ON

Select ON to establish run-time tracing.

OFF

Select OFF to terminate run-time tracing.

CONFIG

Turn internal trace for configuration ON or OFF.

SUBAGENT

Turn internal trace for SNMP subagent ON or OFF.

COMMAND

Turn internal trace for command ON or OFF.

level

Indicates the tracing level to be established. Levels are as follows:

Levels for CONFIG

1	ITRACE for all of config
2	General level of tracing for all of config
3	Tracing for configuration set commands
4	Tracing for configuration get commands
5	Tracing for syslog calls issued by config
100	Tracing for the parser
200	Tracing for scanner
300	Tracing for mainloop
400	Tracing for commands

Levels for SUBAGENT

1	General subagent tracing
2	General subagent tracing plus DPI traces
3	General subagent tracing plus extended storage dump traces
4	All trace levels

Levels for COMMAND

1	ITRACE for all commands

Examples

```
ITRACE ON CONFIG 3
ITRACE OFF SUBAGENT
```

Usage Notes

- This statement is used primarily for diagnostic purposes.
- Trace output goes to SYSPRINT.
- ITRACE ON commands are cumulative until an ITRACE OFF is issued.
- ITRACE should only be set at the direction of an IBM Service representative.

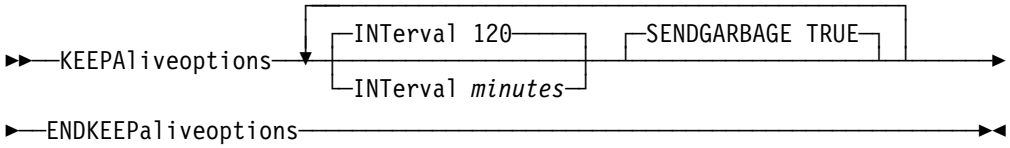
Related Topics

OS/390 TCP/IP OpenEdition Diagnosis Guide

KEEPALIVEOPTIONS Statement

Use the KEEPALIVEOPTIONS statement to specify the operating parameters of the TCP keep-alive packets. The parameters apply to all TCP connections for which keep-alive has been activated through the setsockopt() call of the C socket interface.

Syntax



Parameters

INTERVAL *minutes*

The number of minutes TCP waits after receiving a packet for a connection before it sends a keep-alive packet for that connection. The range is from 0 to 35791 minutes. A value of 0 will disable it.

SENDGARBAGE TRUE

Specifies that the keep-alive packets sent by TCP contain 1 byte of random data and an invalid sequence number, assuring that the data is not accepted by the remote TCP.

Usage Notes

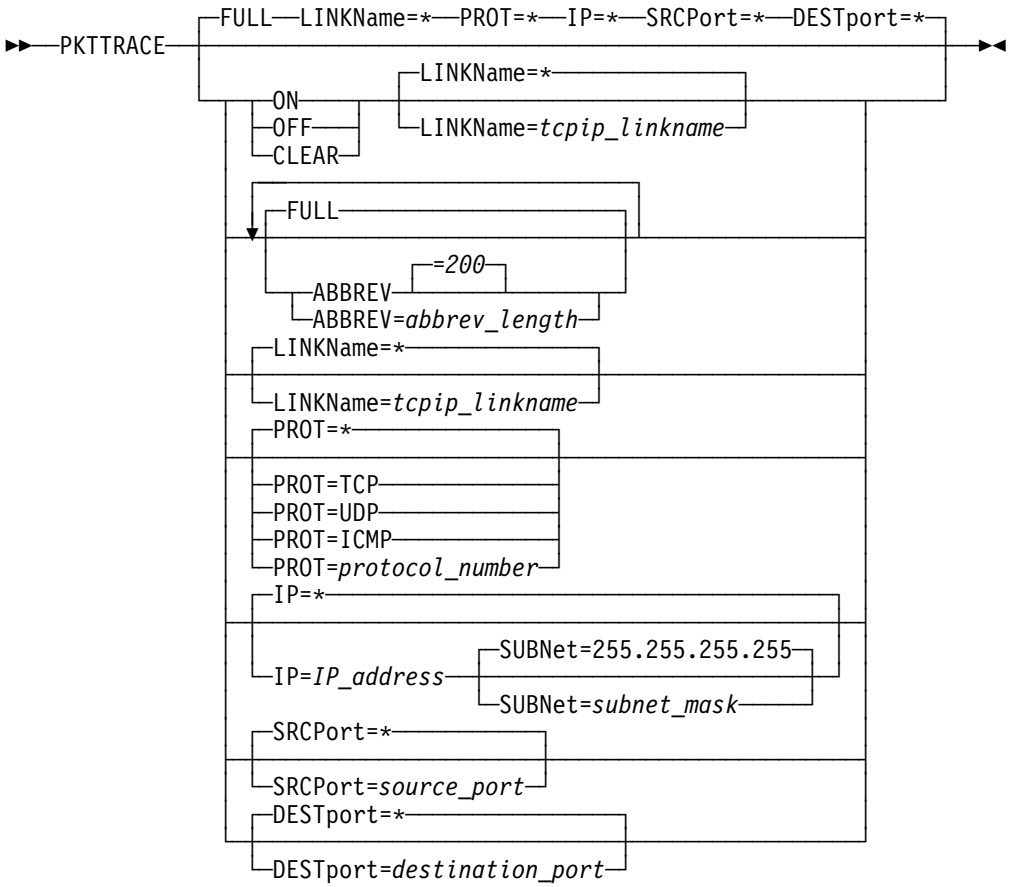
- KEEPALIVE INTERVAL 0 turns off TCP keepalive.
- KEEPALIVE INTERVAL is a global setting for the entire TCP stack. Each socket can select keepalive with setsocketopt().
- The ENDKEEPALIVEOPTIONS statement specifies the end of the KEEPALIVEOPTIONS information. If it is omitted, subsequent entries will generate error messages.

PKTTRACE Statement

Use the PKTTRACE statement to control the packet tracing facility in TCP/IP. You can use this statement to select IP packets as candidates for tracing and subsequent analysis. An IP packet must meet all the conditions specified on the statement for it to be traced.

The PKTTRACE statement consists of two parts. The first part defines to TCP/IP the links that are to be traced and characteristics of how they are to be traced. The second part turns packet tracing ON or OFF, or CLEARs packet trace settings for the links specified on prior PKTTRACE statements or for a single link if the LINKName parameter is used.

Syntax



Parameters

ABBREV

Specifies that a truncated portion of the IP packet is to be traced. You can specify a length between 1 and 65535 or take the default of 200. The ABBREV parameter can be used to reduce the volume of data stored in the trace file.

CLEAR

Disables packet tracing for the links specified and removes the characteristics defining how they should be traced. Use this parameter when tracing is not to be done on the specified link for quite some time.

DESTPORT

Specifies a port number that will be compared with the destination port of inbound and outbound packets. The port number is an integer between 1 and 65535. If the destination port of a packet is the same as the specified port number, the packet will be traced. This comparison is only performed for packets using either the TCP or UDP protocol; packets using other protocols are not traced. If the DESTPORT parameter is omitted, there is no checking of the destination port of packets. If an asterisk (*) is specified, packets of any protocol and any source port will be traced.

FULL

Specifies that the entire IP packet is to be traced.

IP

Specifies an IP address that will be compared with both the source and destination addresses of inbound and outbound packets. If either the source or destination address of a packet matches the specified IP address, the packet will be traced. The IP address must be specified in dotted decimal notation. If the IP option is omitted, or an asterisk (*) is specified, then all IP addresses will be traced.

LINKNAME

Specifies the name of the link defined in the preceding LINK statement. If the LINKNAME parameter is omitted or an asterisk (*) is specified, the PKTTRACE parameters will apply to all links prior to this statement.

To facilitate defining packet tracing when many links are involved, use the PKTTRACE statement with the LINKNAME=* option to define packet tracing characteristics for the majority of the links. Then use individual PKTTRACE statements with specific LINKNAME parameters for each link that must be defined differently from the majority.

OFF

Disables packet tracing for the links specified but retains the characteristics defining how they should be traced. Use this parameter when tracing is not to be done on the specified link but may resume in the near future. By using OFF instead of CLEAR, packet trace definitions do not have to be redefined.

ON

Turns on packet tracing for those links previously defined by one or more PKTTRACE statements, or for a specified link if the LINKName parameter is used.

PROT

Specifies the protocol type to be traced. This can be specified as one of the literals TCP, UDP, or ICMP, or as a number between 0 and 255 (ICMP=1, TCP=6, UDP=17, and RAW=255). If the PROT parameter is omitted or an asterisk (*) is specified, packets of any protocol will be traced.

SRCPORT

Specifies a port number that will be compared with the source port of inbound and outbound packets. The port number is an integer between 1 and 65535. If the source port of a packet is the same as the specified port number, the packet will be traced. This comparison is only performed for packets using either the TCP or UDP protocol; packets using other protocols are not traced. If the SRCPORT parameter is omitted, there is no checking of the source port of

packets. If an asterisk (*) is specified, packets of any protocol and any source port will be traced.

SUBNET
Specifies a subnet mask that applies to the host and network portions of the IP address specified on the accompanying IP parameter. The subnet mask must be specified in dotted decimal notation and must be specified in conjunction with the IP parameter.

Usage Notes

- IP=* implies IP=0.0.0.0 and SUBNET=255.255.255.255.
- The IP address and SUBNET pair specified must be in the same network.
- If a given keyword is specified multiple times, the last value specified is used. If an option appears more than once on a statement, the value associated with the last occurrence of the option is used.
- The Generalized Tracing Facility (GTF) is used to store the trace data in a GTF managed data set. For more information, see *OS/390 V1R3.0 MVS Diagnosis: Tools and Service Aids*. for information on the steps required to perform an IP packet trace.
- The PKTTRACE statement must appear after a valid LINK statement for the link in the PROFILE.TCPIP data set.
- TCP/IP for MVS allows a single TCPIP address space to drive many devices, including more than one instance of any particular device, to provide connections to the TCP/IP network. The PKTTRACE statement supports this capability through the LINKNAME option.
- Options on the statement can appear in any order.
- If no options are specified on the PKTTRACE command, then all packets through all devices will be traced.
- If an error is found while parsing the PKTTRACE statement, an error message is generated, the parameter in error is ignored, and the rest of the statement is parsed. If an error is produced by an incorrect ABBREV value, the ABBREV value is changed to the default.
- Each defined link will have an associated trace profile. The trace profile stores the effective values of each of the trace options for the link. When created, or reset using the CLEAR option, a link's trace profile is set to the default values for the trace options as follows:

  PROT All protocols
  IP All IP addresses
  SUBNET No checking
  SRCPORT No checking
  DESTPORT No checking
  FULL Tracing of the whole IP packet

- More than one PKTTRACE statement can be included in the TCPIP profile or using the VARY TCPIP command. Multiple statements can refer to the same link either by explicitly naming the link or by defaulting to an asterisk ("*"), which indicates all links. When multiple statements are included, the last statement processed will be in affect. If each statement has a specific link name, then the changes only affect those specific links. Otherwise, all links are affected. Parameters not specified will be defaulted. For example, if you specified:

```
PKTTRACE PROT=UDP
PKTTRACE IP=127.0.0.1
PKTTRACE ON
```

Packet trace would be active for all link names, all protocols, all SUBNETs, all SRCPORTs, and all DESTPORTs by default, and for IP address 127.0.0.1. The first PROT=UDP is reset by the default PROT=* on the second PKTTRACE statement. To get both you would have to specify:

```
PKTTRACE PROT=UDP IP=127.0.0.1
PKTTRACE ON
```

PKTTRACE statements that affect all linknames will affect links that have trace turned OFF but have not been cleared.

Examples

The following sample includes several examples of the PKTTRACE statement:

```
; CTC Device and Link
DEVICE CTC1     CTC     D00
LINK   CTCD00   CTC  1  CTC1
;
; CTC Device and Link
DEVICE CTC2     CTC     D02
LINK   CTCD02   CTC  1  CTC2
;
; CTC Device and Link
DEVICE CTC3     CTC     D04
LINK   CTCD04   CTC  1  CTC3
;
; LCS Device and Links
DEVICE LCS1     LCS     100
LINK   TR1      IBMTR        1  LCS1
LINK   LCSC00   ETHERNET     2  LCS1
LINK   LCSF00   FDDI         3  LCS1
;
DEVICE LCS2     LCS     102
LINK   LCS802   802.3        1  LCS2
;
DEVICE LCS3     LCS     104
LINK   LCSE802  ETHEROR802.3 1  LCS3
;
;
; set defaults for all links not specified below
PKTTRACE
; set for CTCD00
PKTTRACE FULL LINKNAME=CTCD00 PROT=* IP=* SRCPORT=* DESTPORT=*
; set for CTCD02
PKTTRACE ABBREV LINKNAME=CTCD02 PROT=TCP IP=9.67.116.124
              SRCPORT=5000 DESTPORT=161
; set for CTCD04
PKTTRACE ABBREV=1 LINKNAME=CTCD04 PROT=UDP IP=9.67.116.124
              SUBNET=255.255.255.255 SRCPORT=161 DESTPORT=5000
; set for TR1
PKTTRACE ABBREV=200 LINKNAME=TR1 PROT=ICMP IP=*
              SRCPORT=5000 DESTPORT=161
; set for LCSC00
PKTTRACE ABBREV=65535 LINKNAME=LCSC00 PROT=1 IP=9.67.116.124
              SUBNET=255.255.255.255 SRCPORT=* DESTPORT=*
; start pkttrace for all definitions above
PKTTRACE ON
; set for LCSF00 not to trace
PKTTRACE OFF LINKNAME=LCSF00
```

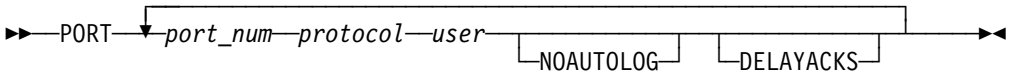
Related Topics

- *OS/390 TCP/IP OpenEdition Diagnosis Guide*

PORT Statement

Use the PORT statement to reserve a port for specified user IDs, procedures, or job names. The PORT statement also specifies the protocol to be used, if the *user* should not be autologged, and if TCP protocol acknowledgements should be delayed.

Syntax



Parameters

port_num

Reserves a port for one or more users. The same port number can appear in more than one PORT statement with different users or more than once in the same PORT statement. This port cannot appear in a range specified by the PORTRANGE statement. If a PORTRANGE statement that includes this port number is specified prior to this statement, this port is ignored. If the PORTRANGE statement follows this statement, then the PORTRANGE statement is ignored. An error message is generated in either case. *port_num* is a value between 1 and 65535.

protocol

Specifies the protocol to be used, either TCP or UDP.

user

For TCP, this specifies one or more user IDs, procedure names, or job names that can use the given port. This allows multiple users to do a passive open on a well-known port and allows multiple servers. When using a procedure name, it must be the member name of the cataloged procedure you use to start the address space, not the name on the EXEC statement in the procedure.

For UDP, only one user ID, procedure name, or job name can be associated with a given port.

For batch jobs which use TCP/IP functions, *user* is the user ID of the batch job as established by USER=*userid* in JCL or through RACF. Use the job name only if there is no user ID associated with the batch job.

NOAUTOLOG

Tells the TCPIP address space ***not*** to restart the server if it was stopped previously. Otherwise, the default is to restart the server if it was stopped previously.

DELAYACKS

Allows you to alter the default TCP/IP behavior for acknowledgements and delay their transmission so that they can be combined with data sent to the foreign host. This affects acknowledgements returned when a packet is received with the PUSH bit on in the TCP header. The default behavior is to return an acknowledgement immediately.

The DELAYACKS parameter on the PORT or PORTRANGE statement only applies to the TCP protocol and only affects acknowledgements on this port connection.

Examples

```
PORT
    20 TCP OMVS             ; OE FTP server
           DELAYACKS        ; Delay transmission acknowledgements
    21 TCP OMVS             ; OE FTP server control port
    23 TCP OMVS             ; OE TELNET server
   111 TCP OMVS             ; OE Portmap server
   111 UDP OMVS             ; OE Portmap server
   161 UDP OSNMPD           ; OE SNMP agent port for SNMP requests
   162 UDP OMVS             ; osnmp command port for receipt of traps
   520 UDP OROUTED          ; OE RouteD server
```

Usage Notes

- Ports defined in a VARY TCPIP,,CMD=OBEYFILE command data set are added to the list of ports already defined. To delete a port, you must use the DELETE statement.
- A port that is not reserved in this list or with the PORTRANGE statement can be used by any user. If you have TCP/IP hosts in your network that use ports in the range 1-1023 for privileged applications, you should reserve them via this statement, the PORTRANGE statement, or the RESTRICTLOWPORTS parameter on the ASSORTEDPARMS, TCPCONFIG, or UDPCONFIG statements.
- For syslogd, you must include the following PORT statement:

  ```
  PORT
      514 UDP OMVS          ; OE syslogd Server
  ```

  This port is required for syslogd to accept log data from remote syslogd servers.
- If you want SNMP ATM Management support, see “Step 4: Configure the ATM Open Systems Adapter 2 (ATM OSA-2) Support” on page 214 for an explanation of specifying the PORT statement.

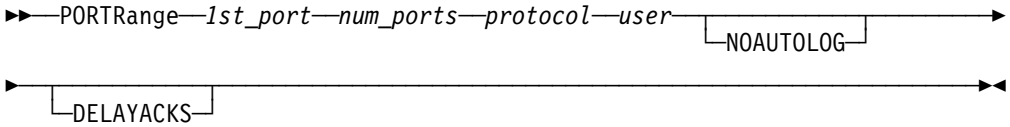
Related Topics

- “DELETE Statement” on page 51
- “GATEWAY Statement” on page 65
- “PORTRANGE Statement”
- “VARY Command—TCPIP Address Space” on page 96

PORTRANGE Statement

Use the PORTRANGE statement to reserve a range of ports for specified user IDs, procedures, or job names. The PORTRANGE statement also specifies the protocol to be used, if the *user* should not be autologged, and if TCP protocol acknowledgements should be delayed.

Syntax



Parameters

1st_port
The starting port for a range of ports to reserve. The same port number can not appear in multiple PORTRANGE statements, nor can the port be specified on both PORTRANGE and PORT statements. If the port is specified on a PORT statement prior to this statement, this port range is ignored. If the port is specified on a PORT statement that follows this statement, the port in the PORT statement is ignored. An error message is generated in either case. *1st_port* is a value between 1 and 65535.

num_ports
The number of ports to reserve. The ports reserved can not overlap other ranges specified by a PORTRANGE statement. No ports within this range can be specified on a PORT statement. If the port is specified on a PORT statement prior to this statement, this port range is ignored. If the port is specified on a PORT statement that follows this statement, the port in the PORT statement is ignored. An error message is generated in either case.

protocol
Specifies the protocol to be used, either TCP or UDP.

user
The user ID for which this port range is reserved. A PORTRANGE can be reserved for one TCP user and one UDP user. Multiple UDP users can never share a port. To have multiple TCP users share ports, use the PORT statement.

For batch jobs which use TCP/IP functions, this field is the user ID of the batch job as established by USER=userid in JCL or through RACF™. Use the job name only if there is no user ID associated with the batch job.

NOAUTOLOG
Tells the TCPIP address space ***not*** to restart the server if it was stopped previously. Otherwise, the default is to restart the server if it was stopped previously.

DELAYACKS
Allows you to alter the default TCP/IP behavior for acknowledgements and delay their transmission so that they can be combined with data sent to the foreign host. This affects acknowledgements returned when a packet is received with the PUSH bit on in the TCP header. The default behavior is to return an acknowledgement immediately.

The DELAYACKS parameter on the PORT or PORTRANGE statement only applies to the TCP protocol and only affects acknowledgements on this port connection.

Examples

This example shows a PORTRANGE statement used to reserve a large number of ports for a single test system.

```
PORTRANGE
    4000 200  TCP TESTSYS
```

This example shows a PORTRANGE statement where port 111 is reserved for both UDP and TCP for one user, and ports 500-504 are reserved for two different users, one using UDP and one using TCP. Note that for multiple users to share the same TCP port, a PORT statement is required. Multiple users cannot share the same UDP port.

```
PORTRANGE
     111   1  UDP  PORTMAP
     111   1  TCP  PORTMAP
     500   5  UDP  USER1
     500   5  TCP  USER2

PORT 600 TCP USER1
     601 TCP USER1
     602 TCP USER1
     600 TCP USER2
     601 TCP USER2
     602 TCP USER2
     600 TCP USER3
     601 TCP USER3
     602 TCP USER3
```

Usage Notes

- A range of ports defined in a VARY TCPIP,,CMD=OBEYFILE command data set are added to the list of ports already defined. To delete a range of ports, you must use the DELETE statement.
- A port that is not reserved by a PORT or PORTRANGE statement can be used by any user. If you have TCP/IP hosts in your network that reserve ports in the range 1-1023 for privileged applications, you should reserve them either via this statement, the PORT statement, or the RESTRICTLOWPORTS parameter on the ASSORTEDPARMS, TCPCONFIG, or UDPCONFIG statements.

Related Topics

- “DELETE Statement” on page 51
- “PORT Statement” on page 83
- “VARY Command—TCPIP Address Space” on page 96

PRIMARYINTERFACE Statement

Use the PRIMARYINTERFACE statement to specify which link to use as the source IP address when communicating with hosts beyond the local network.

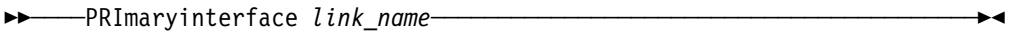
This statement can only be used for one link.

If multiple PRIMARYINTERFACE statements are coded, the last statement that is processed successfully is used.

If a remote host is on one of the networks to which your host is directly attached (such as an Ethernet or Token Ring), TCP/IP will communicate with the remote host using the IP address associated with the interface on that network. However, if a remote host is on a network reachable by one or more routers, TCP/IP will communicate with the remote host using the IP address of the link on the PRIMARYINTERFACE statement.

Note: If PRIMARYINTERFACE statement is not included, TCP/IP will communicate with the remote host using the first IP address listed in the HOME statement.

Syntax



Parameters

link_name
The name of a link as defined in a LINK statement that is to be the primary interface. This link must have been also defined in a previous HOME statement.

Examples

This example shows a PRIMARYINTERFACE statement specifying a token ring:

```
PRIMARYINTERFACE TR1
```

You can verify which HOME entry is primary by using the onetstat —h command:

```
Home address list:
Address          Link             Flg
9.67.113.61      TR1              P
9.67.116.125     CTCD00
127.0.0.1        LOOPBACK
```

Usage Notes

- The PRIMARYINTERFACE statement allows you to designate the IP address of a link other than this first address in the HOME list as the source for communicating with hosts beyond the local network.
- The primary interface is flagged in the onetstat —h display.
- If the PRIMARYINTERFACE statement is not specified, then the first address in the HOME list is the source for communicating with hosts beyond the local network.

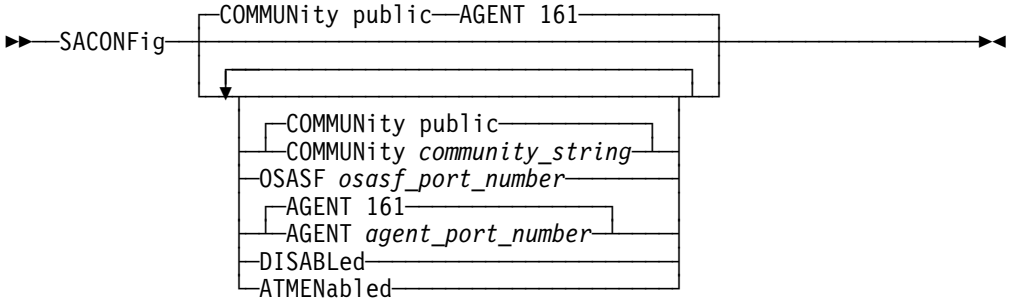
Related Topics

- “DEVICE and LINK Statements” on page 36
- “HOME Statement” on page 71

SACONFIG Statement

Use the SACONFIG statement to configure the SNMP subagent.

Syntax



Parameters

COMMUNITY

A character string of up to 15 characters used as the community name (or password) in establishing contact with the SNMP agent. For the MVS SNMP subagent to communicate with the MVS SNMP agent, the community name specified (or defaulted) on the COMMUNITY keyword must match one that is defined in the PW.SRC dataset used by the SNMP agent or specified (or defaulted) on the -c parameter when the SNMP agent is started. The default value is *public*.

OSASF

A value between 0 and 65535. There is no default. A value of 1 through 65535 indicates a port number and marks the corresponding TCP/IP instance as a candidate to communicate with OSA/SF for retrieval of SNMP management data regarding ATM devices and links. A value of 0 indicates that the corresponding TCP/IP instance is no longer a candidate to communicate with OSA/SF, in the event that the OSA/SF-to-TCP/IP connection is restarted.

When multiple TCP/IP instances specify that ATM management data retrieval is desired, it is recommended that all be configured with the same OSASF parameter. Only one TCP/IP instance connects directly to OSA/SF. Other instances connect to OSA/SF via this primary TCP/IP instance.

AGENT

A port number between 1 and 65535 used in establishing communication with the SNMP agent. For the MVS SNMP subagent to communicate with the MVS SNMP agent, the port number specified must match the port number specified (or defaulted) on the -p parameter when the SNMP agent is started. The default value is 161.

DISABLED

Indicates that the SNMP subagent should not be started. Specify this keyword if little or no SNMP data will be requested from this TCP/IP image. By default, the SNMP subagent is started by TCP/IP initialization.

SNMP variables supported by the MVS SNMP agent will still be available. For information on which MIB variables are supported by the SNMP agent and subagent, see *OS/390 TCP/IP OpenEdition User's Guide*.

ATMENABLED

Indicates that ATM Management support is required at this TCP/IP instance. For optimal performance, specify ATMENABLED only at the instance from which ATM Management support is needed. By default, ATM data retrieval is not enabled.

The SNMP subagent must be enabled, as it provides support for retrieval of SNMP management data about ATM devices and links. Therefore, do not specify the DISABLED parameter for this TCP/IP instance.

To retrieve ATM data, there must also be at least one TCP/IP image running that specifies a value for OSASF. When running multiple TCP/IP instances, it is recommended that all be configured with the same OSASF parameter.

Examples

```
SACONFIG COMMUNITY USACCESS AGENT 528
SACONFIG DISABLED
```

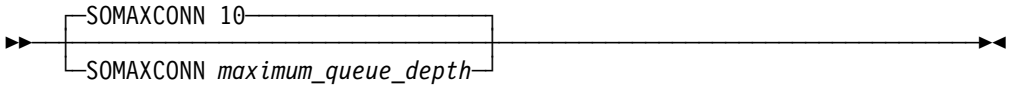
Usage Notes

- If agent_port_number and community string is changed, an SACONFIG statement without specifying AGENT or COMMMUNITY will reset them back to the default.
- If DISABLED is specified, the subagent will NOT be started by TCPIP initialization. If SACONFIG statement itself is NOT specified, the subagent will be started (this is the default).
- The community string 1 to 15 characters and is case sensitive. It is not converted to uppercase by profile processing. It cannot contain any imbedded white space or control characters (such as blank, tab, end of line, or end of file) and cannot contain any imbedded semicolons (semicolons are treated as comment delimiters).

SOMAXCONN Statement

Use the SOMAXCONN statement to specify a maximum length for the connection request queue created by the socket call listen().

Syntax



Parameters

maximum_queue_depth
The maximum number of pending connection requests queued for any listening socket. The default is 10.

Examples

This example shows a SOMAXCONN statement specifying the default number of listening sockets.

```
SOMAXCONN                    10
```

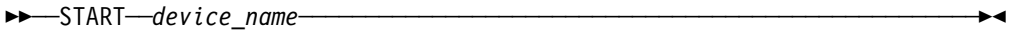
Usage Notes

- This number is stored as a fullword integer, but most implementations of TCP/IP hardcode a value in the range of 5-10.
- This number is the maximum depth for any listening stream socket, but the programmer can specify a shorter queue length on each listen() socket call.
- There is a SOMAXCONN variable in the SOCKET.H file which is hardcoded at 10. If your C socket programs use this variable to determine what the acceptable maximum listen() backlog queue length is, remember to change the header file to reflect the value you specified for TCPIP in SOMAXCONN maximum queue depth.

START Statement

Use the START statement to start a device that is currently stopped. This statement is usually specified at the end of *hlq*.PROFILE.TCPIP.

Syntax



Parameters

device_name
The name of the device to start. This should be the same *device_name* specified in the DEVICE statement.

Examples

This example shows START statements that start devices LCS1 and LCS2.

```
START LCS1
START LCS2
```

Usage Notes

- Each device to be started needs a separate START statement.
- The START statement can also be used in a VARY TCPIP command data set to start:
  - A newly-defined device
  - A device stopped with the STOP statement
  - A device that was never successfully started
- The START statement is not valid for virtual or atm devices. A virtual device is started automatically when a HOME entry is defined to it. It never leaves the started/active state.
- The START and STOP commands are processed *after* all other statements within the profile or obeyfile. Therefore, do not code START and STOP commands for the same device within the same profile or obeyfile, since it cannot be guaranteed that the commands will be processed in the order specified.

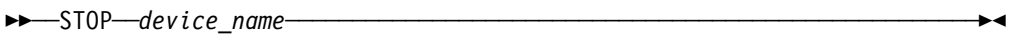
Related Topics

- “DEVICE and LINK Statements” on page 36
- “VARY Command—TCPIP Address Space” on page 96
- “STOP Statement”

STOP Statement

Use the STOP statement in a VARY TCPIP command data set to stop a device that is currently started.

Syntax



Parameters

device_name
The name of the device to be stopped. This should be the same *device_name* specified in the DEVICE statement.

Examples

This example shows STOP statements that stop devices LCS1 and LCS2.

```
STOP LCS1
STOP LCS2
```

Usage Notes

- Storage used by the device driver is not freed; it is reused the next time the device is started.
- An ATM or virtual device can not be stopped.
- The START and STOP commands are processed *after* all other statements within the profile or obeyfile. Therefore, do not code START and STOP com-

mands for the same device within the same profile or obeyfile, since it cannot be guaranteed that the commands will be processed in the order specified.

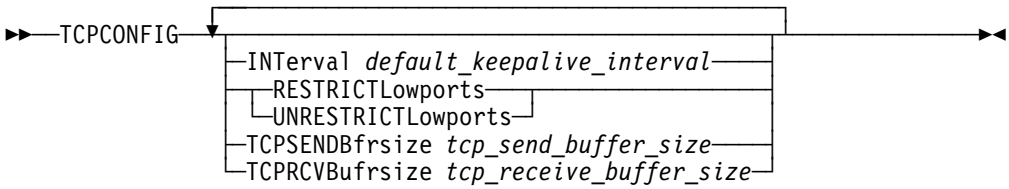
Related Topics

- "DEVICE and LINK Statements" on page 36
- "VARY Command—TCPIP Address Space" on page 96
- "START Statement" on page 91

TCPCONFIG Statement

Use the TCPCONFIG statement to update the TCP layer of TCP/IP.

Syntax



Parameters

INTERVAL *minutes*
The number of minutes TCP waits after receiving a packet for a connection before it sends a keep-alive packet for that connection. The range is from 0 to 35791 minutes. A value of 0 will disable it.

RESTRICTLOWPORTS
When set, ports 1 through 1023 are reserved for users by the PORT and PORTRANGE statements. The RESTRICTLOWPORTS parameter is confirmed by the message:

```
TCP ports 1 thru 1023 are reserved
```

UNRESTRICTLOWPORTS
When set, ports 1 through 1023 are not reserved. The UNRESTRICTLOWPORTS parameter is confirmed by the message:

```
TCP ports 1 thru 1023 are not reserved
```

TCPSENDBFRSIZE *tcp_send_buffer_size*
TCP send buffer size between 256 and 256K. The default is approximately 16384 (16K), but will change slightly with service changes.

TCPRCVBUFRSIZE *tcp_receive_buffer_size*
TCP receive buffer size between 256 and 256K. The default is approximately 16384 (16K), but will change slightly with service changes.

Examples

This example shows a TCPCONFIG statement that reserves ports 1 through 1023 for users by the PORT and PORTRANGE statements:

```
TCPCONFIG RESTRICTLOWPORTS
```

Usage Notes

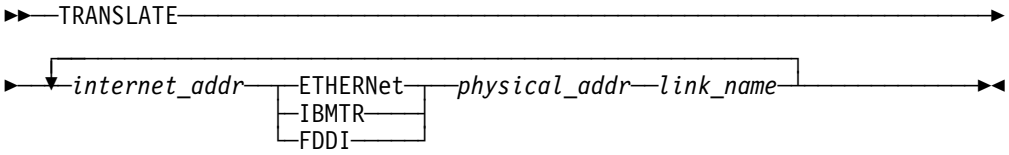
Some of these parameters can be specified on other statements (ASSORTEDPARMS, KEEPALIVEOPTIONS), and the settings from the last statement processed are used. For example, if RESTRICTLOWPORTS is not specified on ASSORTEDPARMS (and thus defaults to OFF) but is specified on a subsequent TCPCONFIG statement, RESTRICTLOWPORTS is set for TCP ports.

TRANSLATE Statement

Use the TRANSLATE statement to indicate the relationship between an internet address and the network address, on a specified link. You can use the TRANSLATE statement, with some limitations, for Ethernet and token-ring hosts that do not support ARP.

The TRANSLATE statement is not valid for virtual devices, or point-to-point devices like CTC.

Syntax



Parameters

internet_addr
The internet address for which a translation is specified.

ETHERNET
Indicates the network address is an Ethernet address.

IBMTR
Indicates the network address is a token-ring address.

FDDI
Indicates the network address is an FDDI address.

physical_addr
The physical address is 6 bytes.

link_name
A network link name (from the LINK statement). The specified *internet_addr* is translated to the specified *net_addr* only when sending on this link. You can include multiple TRANSLATE statement entries for the same *internet_addr* with a different *link_name*.

Examples

This example shows the TRANSLATE statement for FDDI:

```
TRANSLATE
  9.67.51.3    FDDI    FF0000006702   FDDI1
  9.67.22.4    FDDI    FF0000009A05   FDDI1
```

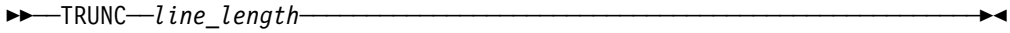
Usage Notes

- Each configuration data set's first executed TRANSLATE statement replaces the internal translation tables (the ARP table), including information dynamically added by ARP, with the new information. Subsequent TRANSLATE statements in the same data set add entries to the table.
- When an incorrect TRANSLATE statement entry is encountered, all entries following that entry on this TRANSLATE statement are ignored. Subsequent TRANSLATE statements in the same profile or obeyfile are processed.
- If the first TRANSLATE statement of a profile contains no internet address or link name, all addresses are removed from the TRANSLATE list.

TRUNC Statement

The TRUNC statement is used to allow sequence numbers in the profile data set.

Syntax



Parameters

line_length The length of the line to be read in a profile. This value can be between 64 and 255. If not specified, the default is 255.

Examples

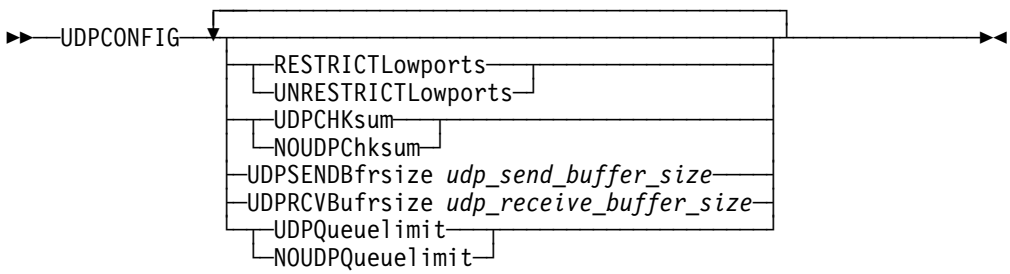
The following example truncates profile input lines after 72 characters:

```
TRUNC 72
```

UDPCONFIG statement

Use the UDPCONFIG statement to update the UDP layer of TCP/IP.

Syntax



Parameters

RESTRICTLOWPORTS
When set, ports 1 through 1023 are reserved for users by the PORT and PORTRANGE statements. The RESTRICTLOWPORTS parameter is confirmed by the message:

```
UDP ports 1 thru 1023 are reserved
```

UNRESTRICTLOWPORTS
Ports 1 through 1023 are not reserved. The UNRESTRICTLOWPORTS parameter is confirmed by the message:

```
UDP ports 1 thru 1023 are not reserved
```

UDPCHKSUM
Used to ensure UDP does check summing.

NOUDPCHKSUM
Used to ensure UDP does not do check summing.

UDPSENDBFRSIZE *udp_send_buffer_size*
Set UDP send buffer size.

UDPRCVBUFRSIZE *udp_receive_buffer_size*
Set UDP receive buffer size.

UDPQUEUELIMIT
Used to set a queue limit for UDP. The UDPQUEUELIMIT parameter is confirmed by the message:

```
A limit on incoming UDP datagram queue set
```

NOUDPQUEUELIMIT
Used to specify that UDP should not have a queue limit. The NOUDPQUEUELIMIT parameter is confirmed by the message:

```
No limit on incoming UDP datagram queue set
```

Examples

This example shows a UDPCONFIG statement that uses check summing, sets no queue limit, and sets the send buffer size to 8192:

```
UDPCONFIG UDPCHK NOUDPQ UDPSENDB 8192
```

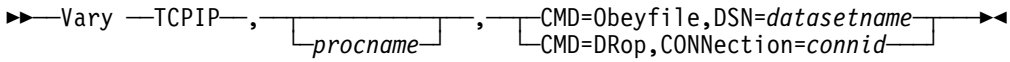
Usage Notes

RESTRICTLOWPORTS can be specified on the ASSORTEDPARMS statement, and the settings from the last statement processed are used. For example, if RESTRICTLOWPORTS is not specified on ASSORTEDPARMS (and thus defaults to OFF) but is specified on a subsequent UDPCONFIG statement, RESTRICTLOWPORTS is set.

VARY Command—TCPIP Address Space

Use the VARY TCPIP command to control some functions of the TCP/IP address space from the operator's console.

Syntax



Parameters

procname
The identifier of the TCP/IP address space. When the *procname* parameter is not specified, there can be only one TCP/IP address space started. If more than one TCP/IP address space is available and no *procname* is specified, the request will fail with an error message.

CMD=OBEYFILE
specify this parameter to make temporary dynamic changes to the system operation and network configuration without stopping and restarting the TCP/IP address space. These changes are in effect until the TCPIP cataloged procedure is started again or until another VARY CMD=OBEYFILE overrides them. Put your changes in the data set specified by the parameter *datasetname*. You can maintain different data sets that contain a subset of the TCP/IP configuration statements and activate them while TCP/IP is running.

DSN=*datasetname*
The parameter DSN= and its value *datasetname* are required after specifying the CMD=OBEYFILE parameter. *datasetname* is the name of a data set containing TCPIP configuration statements. *datasetname* must be a cataloged data set and specified as fully qualified without any quotes. *datasetname* can be either a sequential data set or a member in a PDS.

CMD=DROP
specify this parameter to drop a connection.

CONNECTION= *connid*
The parameter CONNECTION= and its value *connid* are required after specifying the CMD=DROP parameter. *connid* is the connection identifier for the TCPIP socket connection that is to be dropped. Issue the onetstat —c command to obtain the connection identifier for the TCP/IP socket connection that you want to drop.

Examples

The first set of examples are for updating system operation and network configuration information without stopping and restarting the TCP/IP address space.

1. The first example is directed to a TCPIP address space started by the identifier TCPV3R3 and assumes the sequential data set USER99.TCPIP.OBEYFIL1 contains TCP/IP configuration statements:

   ```
   VARY TCPIP,TCPV3R3,CMD=OBEYFILE,DSN=USER99.TCPIP.OBEYFIL1
   ```

2. The next example assumes there is only one TCPIP address space and that OBEYFIL2 is a member of the PDS USER99.TCPIP and contains TCP/IP configuration statements:

   ```
   VARY TCPIP,,CMD=O,DSN=USER99.TCPIP(OBEYFIL2)
   ```

The next set are examples of dropping TCP/IP socket connections.

1. The first example is directed to a TCPIP address space started by the identifier TCPV3R3 and demonstrates how to drop a TCP connection number 5001:

   ```
   VARY TCPIP,TCPV3R3,CMD=DROP,CONNECTION=5001
   ```

2. The next example assumes there is only one TCPIP address space and demonstrates how to drop a UDP connection number 6001:

   ```
   VARY TCPIP,,CMD=DROP,CONNECTION=6001
   ```

Usage Notes

1. Authorization is through the user's RACF profile containing the MVS.VARY.TCPIP.OBEYFILE definition for CMD=OBEYFILE and the MVS.VARY.TCPIP.DROP definition for CMD=DROP.
2. The DSN= parameter cannot be an HFS file.

Chapter 4. Defining the TCP/IP Client System Parameters

Before You Configure...:

> Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

This chapter describes how you can define the TCP/IP system parameters required by the client programs. These parameters are specified using the configuration statements in a TCPIP.DATA file. For further information, see *OS/390 OpenEdition Planning*. Also see the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

Configuration Process

You must configure a file containing TCPIP.DATA statements. The OpenEdition search path for a file that contains these statements is:

- Environment variable RESOLVER_CONFIG
- The HFS file /etc/resolv.conf
- Non-OE search path.

The non-OE search path varies depending on the execution environment of the application needing access to a TCPIP.DATA file. For help in setting up multiple copies of TCPIP.DATA, see “Considerations for Multiple Instances of TCP/IP” on page 26. A single TCP/IP copy (or instance) needs a TCPIP.DATA file for the instance and all local applications that use the file to be configured correctly. Remember that the environment variable, RESOLVER_CONFIG, and the HFS file, /etc/resolv.conf, both fall before the non-OE search path. If either of these exist, the first one that is found is searched for the needed statement. The search does not continue if the statement is not found.

Create a TCPIP.DATA file by copying the sample provided in TCPIP.SEZAINST(TCPDATA) and modifying it to suit your local conditions.

Allocate this data set with either sequential (PS) or partitioned (PO) organization, a fixed block format (FB), a logical record length (LRECL) of 80, and any valid blocksize value for a fixed block, such as 3120. This file can also be the HFS file /etc/resolv.conf, or an HFS file that is pointed to by either the environment variable RESOLVER_CONFIG or the SYSTCPD DD in a JCL procedure. The environment variable RESOLVER_CONFIG can also point to an MVS data set or PDS.

You can use any name for the TCPIP.DATA data set if you access it using the //SYSTCPD DD statement, or use ENVAR to set RESOLVER_CONFIG,in the JCL for all the servers, logon procedures, and batch jobs that execute TCP/IP functions. If you are not using the //SYSTCPD DD statement, the environment variable, or /etc/resolv.conf,then the data set name must conform to the conventions described in “Information Specific to Data Sets in Configuration File Search Orders” on page 6. Another alternative is to use the well-known data set name SYS1.TCPPARMS(TCPDATA). You can issue the HOMETEST command to verify the actual name of the data set name the system finds for TCPIP.DATA.

Note: If either the environment variable RESOLVER_CONFIG or /etc/resolv.conf is being used, then HOMETEST is of no use since OE extensions to the traditional TCP/IP search path will not be understood.

For more information on resolver configuration files, see the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

Each configuration statement can be preceded by an optional *system_name*. This permits configuration information for multiple systems to be specified in a single *hlq*.TCPIP.DATA data set. The *system_name* is matched against the name of the system on which you are running. The name of the system is taken from the node name in the IEFSSNxx member of PARMLIB.

The statements are processed in the order they appear in the data set. The following rules apply to this processing.

- If the *system_name* does not match the name of the system, the configuration statement is ignored.
- If *system_name* is blank, the configuration statement is in effect on every system.
- If the *system_name* matches the node name, the configuration statement that follows it is in effect.
- The ***last*** statement that matches, is effective.

For example, if you have the following three TCPIPJOBNAME statements, MVS6 would look for a TCPIP cataloged procedure named TCPBTA2, MVSA would look for TCPV3, and all other systems would look for TCPMCWN.

```
      TCPIPJOBNAME TCPMCWN
MVS6: TCPIPJOBNAME TCPBTA2
MVSA: TCPIPJOBNAME TCPV3
```

But if you reversed the order, all systems would try to find the procedure named TCPMCWN.

```
MVS6: TCPIPJOBNAME TCPBTA2
MVSA: TCPIPJOBNAME TCPV3
      TCPIPJOBNAME TCPMCWN
```

Summary of Statements in TCPIP.DATA

The statements and system parameters are summarized in Table 7.

Table 7 (Page 1 of 2). Summary of TCPIP.DATA Configuration Statements

Statement	Description	Page
ALWAYSWTO	Issue WTO messages for all servers	104
DATASETPREFIX	Set the high-level qualifier for dynamic allocation of data sets	104
DOMAINORIGIN	Specify the domain origin that is appended to the host name to form the fully qualified domain name for a host	105
HOSTNAME	Specify the TCP host name of the OS/390 server	106
LOADDBCSTABLES	Tell FTP which DBCS translation tables can be loaded	106
MESSAGECASE	Specify case translation for the FTP server and osnmpd	108

Table 7 (Page 2 of 2). Summary of TCPIP.DATA Configuration Statements

Statement	Description	Page
NSINTERADDR	Define the IP address of a name server in dotted decimal format	109
NSPORTADDR	Specify the name server port number	110
RESOLVEVIA	Specify the protocol used by the resolver to communicate with the name server	110
RESOLVERTIMEOUT	Specify how long the resolver waits for a response while trying to communicate with the name server	111
RESOLVERUDPRETRIES	Specify how many times the resolver tries to connect to the name server with when using UDP datagrams	112
TCPIPJOBNAME	Specify the member name of the cataloged procedure used to start the TCPIP address space	112
TRACE RESOLVER	Trace all queries to and responses from the name server	113

Note: Because OS/390 OpenEdition TCP/IP currently supports only OpenEdition sockets, the SOCKBULKMODE, SOCKDEBUG, SOCKDEBUGBULKPERF0, and SOCKNOTESTSTOR statements are not supported and will not be used if specified.

Sample TCPIP.DATA Data Set (TCPDATA)

The following sample is used to specify configuration information of client parameters.

```
;
;*********************************************************************
;                                                                    *
;   Name of Data Set:     TCPIP.DATA                                 *
;                                                                    *
;   COPYRIGHT = NONE.                                                *
;                                                                    *
;   This data, TCPIP.DATA, is used to specify configuration          *
;   information required by TCP/IP client programs.                  *
;                                                                    *
;                                                                    *
;   Syntax Rules for the TCPIP.DATA configuration data set:          *
;                                                                    *
;   (a) All characters to the right of and including a ; will be     *
;       treated as a comment.                                        *
;                                                                    *
;   (b) Blanks and <end-of-line> are used to delimit tokens.         *
;                                                                    *
;   (c) The format for each configuration statement is:              *
;                                                                    *
;       <SystemName||':'>  keyword  value                            *
;                                                                    *
;       where <SystemName||':'> is an optional label that can be     *
;       specified before a keyword; if present, then the keyword-    *
;       value pair will only be recognized if the SystemName matches *
;       the node name of the system, as defined in the IEFSSNxx      *
;       PARMLIB member.  This optional label permits configuration   *
;       information for multiple systems to be specified in a single *
;       TCPIP.DATA data set.                                         *
```

```
;                                                                     *
;       NOTE:  You should define the SystemName in the IEFSSNxx       *
;              PARMLIB member to be the same as your JES2 or JES3     *
;              node name.  This is required for correct delivery of   *
;              SMTP mail.                                             *
;                                                                     *
;*********************************************************************
; TCPIPJOBNAME specifies the name of the started procedure that was
; used to start the TCPIP address space.    TCPIP is the default.
;
TCPIPJOBNAME TCPIP
;
; HOSTNAME specifies the TCP host name of this system.  If not
; specified, the default HOSTNAME will be the node name specified
; in the IEFSSNxx PARMLIB member.
;
; For example, if this TCPIP.DATA data set is shared between 2
; systems, OURMVSNAME and YOURMVSNAME, then the following 2 lines
; will define the HOSTNAME correctly on each system.
;
OURMVSNAME:    HOSTNAME  OURTCPNAME
YOURMVSNAME:   HOSTNAME  YOURTCPNAME
;
; DOMAINORIGIN specifies the domain origin that will be appended
; to host names passed to the resolver.  If a host name contains
; any dots, then the DOMAINORIGIN will not be appended to the
; host name.
;
DOMAINORIGIN  YOUR.DOMAIN.NAME
;
; NSINTERADDR specifies the IP address of the name server.
; LOOPBACK (14.0.0.0) specifies your local name server.  If a name
; server will not be used, then do not code an NSINTERADDR statement.
; (Comment out the NSINTERADDR line below).  This will cause all names
; to be resolved via site table lookup.
;
NSINTERADDR  14.0.0.0
;
; NSPORTADDR specifies the foreign port of the name server.
; 53 is the default value.
;
NSPORTADDR 53
;
; RESOLVEVIA specifies how the resolver is to communicate with the
; name server.  TCP indicates use of TCP virtual circuits.  UDP
; indicates use of UDP datagrams.  The default is UDP.
;
RESOLVEVIA UDP
;
; RESOLVERTIMEOUT specifies the time in seconds that the resolver
; will wait to complete an open to the name server (either UDP or TCP).
; The default is 30 seconds.
;
RESOLVERTIMEOUT 30
;
; RESOLVERUDPRETRIES specifies the number of times the resolver
; should try to connect to the name server when using UDP datagrams.
; The default is 1.
```

```
;
RESOLVERUDPRETRIES 1
;
; TRACE RESOLVER will cause a complete trace of all queries to and
; responses from the name server or site tables to be written to
; the user's console.  This command is for debugging purposes only.
;
; TRACE RESOLVER
;
;
; You can specify DATASETPREFIX in the PROFILE.TCPIP and TCPIP.DATA
; data sets.  The character string specified as a parameter on
; DATASETPREFIX takes precedence over both the distributed or modified
; data set prefix name as changed by the EZAPPRFX installation job.
; If this statement is used in a profile or configuration
; data set that is allocated to a client or a server, then
; that client or server dynamically allocates additional required data
; sets using the value specified for DATASETPREFIX as the data set name
; prefix.  The DATASETPREFIX parameter can be up to 26 characters long
; and the parameter must NOT end with a period.
;
; For more information please see "Understanding TCP/IP Data Set
; Names" in the Customization and Administration Guide.
;
DATASETPREFIX TCPIP;
;
; MESSAGECASE MIXED indicates to the FTP server, FTP client, TELNET
; client, and PING client that all messages should be displayed in
; mixed case.  MESSAGECASE UPPER indicates to the FTP server, FTP
; client, TELNET client, and PING client that all messages should
; be displayed in uppercase.  Mixed case inserts in messages will
; will not be uppercased.
; If MESSAGECASE is not specified, mixed case messages will be used.
;
; MESSAGECASE MIXED
; MESSAGECASE UPPER
;
; LOADDBCSTABLES indicates to the FTP server and FTP client which DBCS
; translation tables should be loaded at initialization time. Remove
; from the list any tables that are not required. If LOADDBCSTABLES is
; not specified, no DBCS tables will be loaded.
;
; LOADDBCSTABLES JIS78KJ JIS83KJ SJISKANJI EUCKANJI HANGEUL KSC5601
; LOADDBCSTABLES TCHINESE BIG5 SCHINESE;
;
; End of file.
;
```

TCPIP.DATA Configuration Statements

This section explains each statement for the TCPIP.DATA data set in detail.

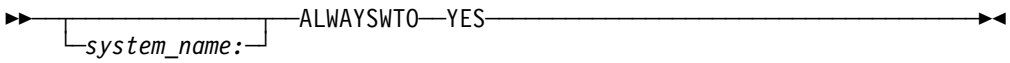
Syntax Conventions

Within *hlq*.TCPIP.DATA, blanks and record boundaries are used to separate tokens. All characters to the right of, and including, a semicolon are treated as comments.

ALWAYSWTO Statement

Use the ALWAYSWTO statement to have TCP/IP issue WTO messages for some servers.

Syntax



Parameters

system_name:
: The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*nodename*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

YES
: Indicates that all server messages are to be displayed on the console.

Examples

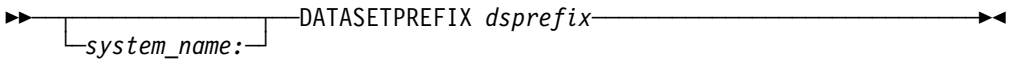
Have TCP/IP send the WTO messages on the MVSMFG2 system.

```
MVSMFG2:ALWAYSWTO YES
```

DATASETPREFIX Statement

Use the DATASETPREFIX statement to set the high-level qualifier for the dynamic allocation of data sets in TCP/IP.

Syntax



Parameters

system_name:
: The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*nodename*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

dsprefix
: The prefix to use as the high-level qualifier for the dynamic allocation of data sets. The default high-level qualifier distributed with the system is TCPIP.

Examples

Set the data set prefix for all client and server datasets:

```
DATASETPREFIX TCPIP.V3R2
```

Usage Notes

The DATASETPREFIX in TCPIP.DATA is used by all clients and all servers except the TCPIP address space. Telnet first uses the DATASETPREFIX statement in TCPIP.DATA. If it does not find one, it uses the DATASETPREFIX statement in PROFILE.TCPIP.

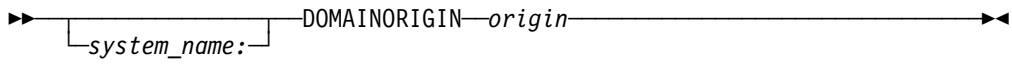
Related Topics

"Configuration Data Sets and HFS Files" on page 5

DOMAINORIGIN Statement

Use the DOMAINORIGIN statement to specify the domain origin that is appended to the host name to form the fully qualified domain name for a host.

Syntax



Parameters

system_name:
: The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*nodename*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

origin
: The domain origin that is appended to the host name. This name cannot not have imbedded dots.

Examples

- This example appends CASTLE as the domain origin for hosts on the MVSMFG4 system:

  ```
  MVSMFG4: DOMAINORIGIN  CASTLE
  ```

- This example does not append the domain origin of BOBS.YOUR.UNCLE to the host name:

  ```
  DOMAINORIGIN  BOBS.YOUR.UNCLE
  ```

Usage Notes

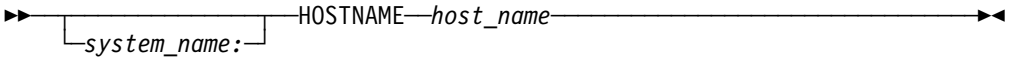
- No case translation is performed on the domain origin.
- If the resolver is passed a host name that does not contain any dots (in dotted decimal notation), the domain origin is appended to the host name. If the host name passed to the resolver contains dots, the domain origin is not appended to the host name.

- The DOMAINORIGIN configuration statement must be customized at each site.

HOSTNAME Statement

Use the HOSTNAME statement to specify the TCP host name of this OS/390 server. The fully qualified domain name for the host is formed by concatenating this host name with the domain origin (specified by the DOMAINORIGIN configuration statement).

Syntax



Parameters

system_name:
: The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*`nodename`*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

host_name
: The host name. If not specified, defaults to the *nodename* specified in the IEFSSNxx PARMLIB member.

Examples

The TCPIP.DATA data set will be shared between 2 systems, MVSMFG4 and MVSADM1. The HOSTNAME statements define the host name on each system.

```
MVSMFG4: HOSTNAME MVSMFG4
MVSADM1: HOSTNAME MVSADM1
```

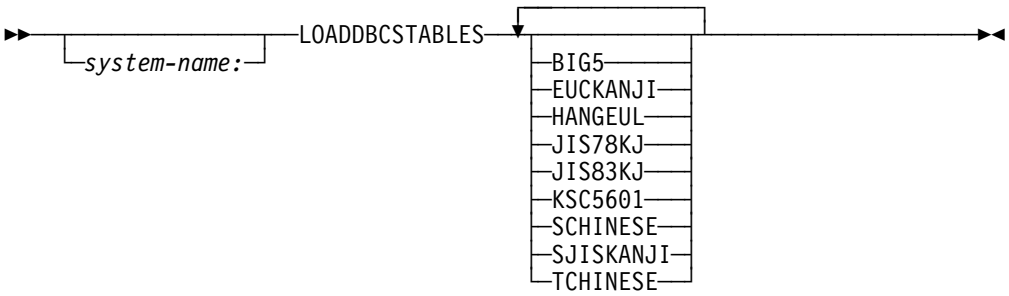
Usage Notes

Case translation is not performed on the host name.

LOADDBCSTABLES Statement

Use the LOADDBCSTABLES statement to tell the FTP server and client which DBCS translation tables can be loaded.

Syntax



Parameters

system_name:

The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*nodename*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

BIG5

Indicates to the FTP server and client that the BIG5 DBCS translation table should be loaded from the TCPCHBIN binary translate table data set.

EUCKANJI

Indicates to the FTP server and client that the Extended Unix Code Kanji DBCS translation table should be loaded from the TCPKJBIN binary translate table data set.

HANGEUL

Indicates to the FTP server and client that the Hangeul DBCS translation table should be loaded from the TCPHGBIN binary translate table data set.

JIS78KJ

Indicates to the FTP server and client that the JIS 1978 Kanji DBCS translation table should be loaded from the TCPKJBIN binary translate table data set.

JIS83KJ

Indicates to the FTP server and client that the JIS 1983 Kanji DBCS translation table should be loaded from the TCPKJBIN binary translate table data set.

KSC5601

Indicates to the FTP server and client that the Korean Standard Code KSC-5601 DBCS translation table should be loaded from the TCPHGBIN binary translate table data set.

SCHINESE

Indicates to the FTP server and client that the Simplified Chinese DBCS translation table should be loaded from the TCPSCBIN binary translate table data set.

SJISKANJI

Indicates to the FTP server and client that the Shift JIS Kanji DBCS translation table should be loaded from the TCPKJBIN binary translate table data set.

TCHINESE

Indicates to the FTP server and client that the Traditional Chinese (5550) DBCS translation table should be loaded from the TCPCHBIN binary translate table data set.

Examples

Load the Korean Standard Code KSC-5601 and the Traditional Chinese (5550) DBCS translation tables:

```
LOADDBCSTABLES KSC5601 TCHINESE
```

Usage Notes

- You can select any or all the of translation tables or specify none. However, additional virtual storage may be required by the FTP server and client when a large number of translation tables are loaded at the same time.
- All the parameters must fit one line. You can repeat the LOADDBCSTABLES statement as necessary to specify additional tables to be loaded.
- If the LOADDBCSTABLES parameter is not specified, is specified incorrectly, or if *hlq*.TCPIP.DATA is not accessible, then no DBCS translation tables will be loaded, and the corresponding FTP server and client DBCS transfer types will be unavailable.
- The IBMKANJI transfer type does not require any translation table to be loaded.

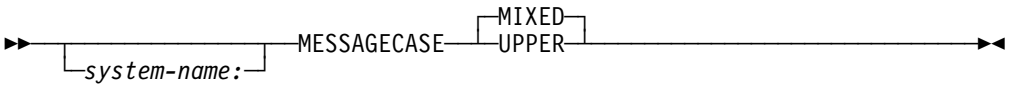
Related Topics

"Using Translation Tables" in the TCP/IP for MVS: Customization and Administration Guide.

MESSAGECASE Statement

Use the MESSAGECASE statement to specify whether to convert Write To Operator (WTO) messages into uppercase for the FTP server and the osnmpd command.

Syntax



Parameters

system_name:

The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*nodename*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

MIXED

Indicates to the FTP server and the osnmpd command that all WTO messages should be displayed in mixed case.

UPPER

Indicates to the FTP server and the osnmpd command that all WTO messages should be displayed in uppercase.

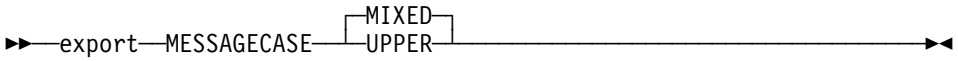
Examples

Display all messages to the MVSTEST system in upper case:

```
MVSTEST: MESSAGECASE UPPER
```

Usage Notes

- If you specify MIXED, no case conversion is performed on WTO messages.
- If the MESSAGECASE statement is not specified, is specified incorrectly, if MIXED or UPPER are not specified, or if *hlq*.TCPIP.DATA is not accessible, then mixed case messages will be displayed.
- Any WTO messages that are displayed by the FTP server and the osnmpd agent at initialization, ***before*** *hlq*.TCPIP.DATA is read, will be displayed in uppercase.
- All WTO messages issued by the TCPIP stack will be displayed in upper case and are not affected by the MESSAGECASE value.
- Additionally, the MESSAGECASE statement can be set from the OpenEdition MVS shell environment by exporting the MESSAGECASE environment variable.

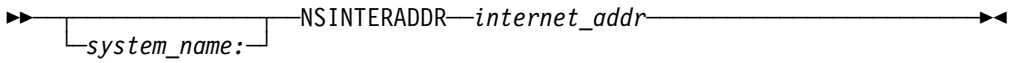


  The setting of the MESSAGECASE environment variable overrides any setting found in TCPIP.DATA. If MESSAGECASE is not defined as a environment variable or as a statement in TCPIP.DATA, the WTO message will remain in mixed case.

NSINTERADDR Statement

Use the NSINTERADDR statement to define the IP address of a name server in dotted decimal format.

Syntax



Parameters

system_name:
: The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*`nodename`*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

internet_addr
: The IP address of a name server.

Examples

Specify the IP address of the name server to be 14.13.12.11:

```
NSINTERADDR  14.13.12.11
```

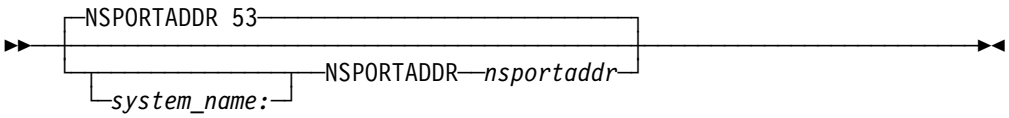
Usage Notes

- You can repeat this statement as many times as you need to specify the IP addresses of alternative name servers.
- Connections to the name servers are attempted in the order they appear in the *hlq*.TCPIP.DATA data set.
- If no NSINTERADDR statements are coded in the *hlq*.TCPIP.DATA data set, the resolver looks for all domain names in the site table, and does not attempt to use a name server.

NSPORTADDR Statement

Use the NSPORTADDR statement to specify the name server port number.

Syntax



Parameters

system_name:
The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*nodename*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

nsportaddr
The name server port number. The default is port 53.

Examples

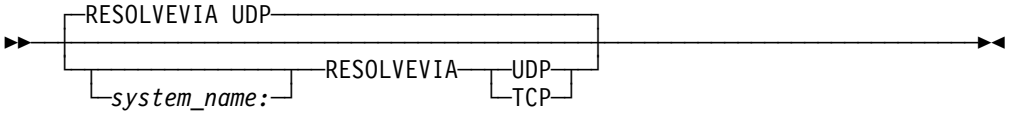
Specify the foreign port of the name server to be 55:

```
NSPORTADDR 55
```

RESOLVEVIA Statement

Use the RESOLVEVIA statement to specify the protocol used by the resolver to communicate with the name server.

Syntax



Parameters

system_name:
The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*`nodename`*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

UDP
Specifies that the protocol is UDP. The default protocol is UDP.

TCP
Specifies that the protocol is TCP.

Examples

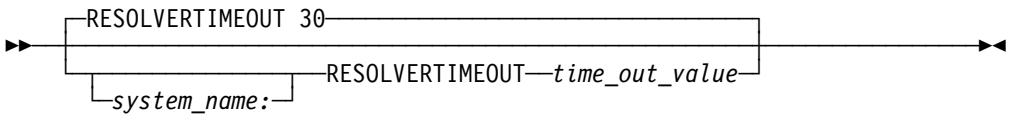
Specify that the resolver is to communicate with the name server using TCP virtual circuits:

```
RESOLVEVIA TCP
```

RESOLVERTIMEOUT Statement

Use the RESOLVERTIMEOUT statement to specify the number of seconds the resolver waits for a response while trying to communicate with the name server (either UDP or TCP).

Syntax



Parameters

system_name:
The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*`nodename`*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

time_out_value
The number of seconds the resolver waits until a response is received. The default open time-out is 30 seconds.

Examples

Specify a 10 second waiting time for the resolver when completing an open to the name server:

```
RESOLVERTIMEOUT 10
```

RESOLVERUDPRETRIES Statement

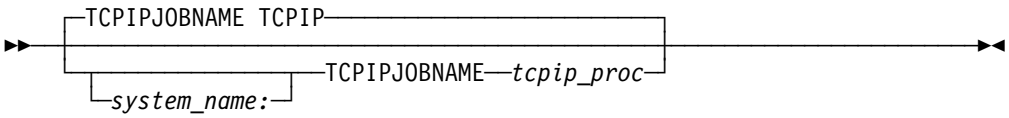
Defaults

REQTEXT

TCPIPJOBNAME Statement

Use the TCPIPJOBNAME statement to specify the member name of the procedure used to start the TCPIP address space.

Syntax



Parameters

system_name:
: The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*`nodename`*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

tcpip_proc
: The name of the member in the cataloged procedure library that is used to start the TCPIP address space. The default is TCPIP.

Examples

Specify TCPIP33 as the name of the procedure that was used to start the TCPIP address space:

```
TCPIPJOBNAME TCPIP33
```

Usage Notes

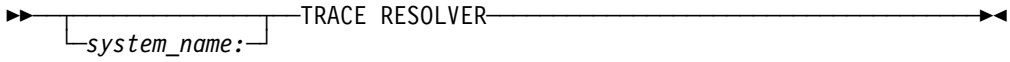
You must specify the proper data set name of the TCPIP address space on your system. If *tcpip_proc* is not the name of the started TCPIP address space, clients will fail at startup with an irrecoverable interaddress communication error.

For more information about why the TCPIPJOBNAME parameter must match the name of the associated TCP/IP address space and be the same name as that defined for the corresponding AF_INET physical file system in the BPXPRMxx member used to configure OpenEdition, see “Considerations for Multiple Instances of TCP/IP” on page 26.

TRACE RESOLVER Statement

Use the TRACE RESOLVER statement to have a complete trace of all queries to and responses from the name server to be written to the user's console.

Syntax



Parameters

system_name:

The system name is derived from the line containing the definition, `VMCF,MVPXSSI,`*`nodename`*`,` in the IEFSSNxx member of PARMLIB. This parameter should be set to the same name as your JES NJE *nodename*. The colon is required.

Examples

Do a complete trace of all queries to and from the name server:

```
TRACE RESOLVER
```

Usage Notes

The TRACE RESOLVER statement is used for debugging purposes only.

Chapter 5. Configuring the Site Table

Before You Configure...:

> Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.
>
> Also, certain socket calls related to name and address resolution are affected by the OpenEdition search path and resolution process. For more information, see “Understanding TCP/IP Data Set Names with OpenEdition MVS” in *OS/390 OpenEdition Planning*. Also see the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

The site table is generated from the *hlq*.HOSTS.LOCAL data set. This data set contains descriptions of local host entries in the HOSTS format. A sample HOSTS.LOCAL data set is created during installation.

This chapter describes how to update the sample *hlq*.HOSTS.LOCAL data set and use it to generate the two data sets, *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO, which function as your site table. It also explains how to test the site table when you are done.

To insure that the TCPIP.DATA statements that affect the resolution services are properly defined for your configuration, see Chapter 4, “Defining the TCP/IP Client System Parameters” on page 99.

The OpenEdition search path for HOSTS.SITEINFO configuration is:

- Value of the environment variable X_SITE (if set).

  This environment variable can contain the information that, in a non-OE environment, was contained in a HOSTS.SITEINFO file that was created by MAKESITE.

- The HFS file /etc/hosts (if it exists).

  This is the equivalent, default, OpenEdition file that can be used in place of a HOSTS.SITEINFO file.

- userid.HOSTS.SITEINFO or jobname.HOSTS.SITEINFO
- tcpip.HOSTS.SITEINFO

  tcpip represents the value of the DATASETPREFIX statement in a TCPIP.DATA file. The default is TCPIP.

HOSTS.SITEINFO information is used by the following functions:

- gethostbyname()
- sethostent()
- gethostent()
- endhostent()
- getnetbyname()

The OpenEdition search path for HOSTS.ADDRINFO configuration is:

- Value of the environment variable X_ADDR (if set).

  This environment variable can contain the information that, in a non-OE environment, was contained in a HOSTS.ADDRINFO file that was created by MAKESITE.

- The HFS file /etc/hosts (if it exists).

  This is the equivalent, default OpenEdition file that can be used in place of a HOSTS.SITEINFO file.

- userid.HOSTS.SITEINFO or jobname.HOSTS.SITEINFO
- tcpip.HOSTS.SITEINFO

  tcpip represents the value of the DATASETPREFIX statement in a TCPIP.DATA file. The default is TCPIP.

HOSTS.ADDRINFO information is used by the following functions:

- getnetbyaddr()
- setnetent()
- getnetent()
- endnetent()
- gethostbyaddr()

Note that if /etc/hosts exists, it overrides both a HOSTS.SITEINFO data set and a HOSTS.ADDRINFO data set since it contains both types of configuration.

Configuration Process

Note: If you are not using the OpenEdition environment variables X_SITE and X_ADDR, or the HFS file /etc/hosts, use the following steps to configure your site table:

1. Update the HOSTS.LOCAL data set
2. Run MAKESITE

Step 1: Update the HOSTS.LOCAL Data Set

A sample HOSTS.LOCAL data set is distributed with TCP/IP for MVS. The installation job, EZAGETIN, copies this data set from *hlq*.SEZAINST(HOSTS) to *hlq*.HOSTS.LOCAL for you. Because each site is unique and requires customized statements, you should only use this data set as a guideline. Update the HOST, NET, and GATEWAY entries in this data set to suit your installation.

HOST Entries

One line of the *hlq*.HOSTS.LOCAL data set is used for each distinct host and ends with 4 colons. Each host can have multiple IP addresses and multiple names. The line for each host has 3 essential fields, separated by colons. These fields are:

- The keyword *HOST*
- A list, separated by commas, of IP addresses for that host
- A list, separated by commas, of fully qualified names for that host

For example, if you have 2 local hosts, LOCAL1 (IP addresses 192.6.77.4 and 192.8.4.1) and LOCAL2 (with an alias LOCALB and IP address 192.6.77.2), append the following lines to the *hlq*.HOSTS.LOCAL data set:

```
HOST : 192.6.77.4, 192.8.4.1 : LOCAL1 ::::
HOST : 192.6.77.2 : LOCAL2, LOCALB ::::
```

Note: The maximum length for a host allowed in the HOST tables is 24 characters. However, the name server does not have a maximum character length.

NET and GATEWAY Entries

The NET and GATEWAY statements are not used by TCP/IP for MVS applications. However, some socket calls require the NET entries. If your programs do not need the NET and GATEWAY statements, delete them before invoking MAKESITE.

Sample HOSTS.LOCAL Data Set (HOSTS)

Following is the sample HOSTS.LOCAL data set:

```
; HOSTS.LOCAL
; -----------
; COPYRIGHT = NONE.
;
; The format of this file is documented in RFC 952, "DoD Internet
; Host Table Specification".
;
; The format for entries is:
;
; NET : ADDR : NETNAME :
; GATEWAY : ADDR, ALT-ADDR : HOSTNAME : CPUTYPE : OPSYS : PROTOCOLS :
; HOST : ADDR, ALT-ADDR : HOSTNAME, NICKNAME : CPUTYPE : OPSYS : PROTOCOLS :
;
; Where:
;   ADDR, ALT-ADDR = IP address in decimal, e.g., 26.0.0.73
;   HOSTNAME, NICKNAME = the fully qualified host name and any nicknames
;   CPUTYPE = machine type (PDP-11/70, VAX-11/780, IBM-3090, C/30, etc.)
;   OPSYS = operating system (UNIX, TOPS20, TENEX, VM/SP, etc.)
;   PROTOCOLS = transport/service (TCP/TELNET,TCP/FTP, etc.)
;   : (colon) = field delimiter
;   :: (2 colons) = null field
; *** CPUTYPE, OPSYS, and PROTOCOLS are optional fields.
;
;   MAKESITE does not allow continuation lines, as described in
;   note 2 of the section "GRAMMATICAL HOST TABLE SPECIFICATION"
;   in RFC 952.  Entries should be specified on a single line of
;   up to a maximum of 512 characters per line.
;
;
; Note:  The NET and GATEWAY statements are not used by the TCP/IP for
;        MVS applications.  However, some socket calls require the NET
;        entries.  For better performance, if your programs do not need
;        the NET and GATEWAY statements, delete them before running
;        the MAKESITE program.
;
;
HOST : 9.67.43.100 : NAMESERVER ::::
HOST : 9.67.43.126 : RALEIGH ::::
HOST : 129.34.128.245, 129.34.128.246 : YORKTOWN, WATSON ::::
```

```
;
NET  : 9.67.43.0 : RALEIGH.IBM.COM :
;
GATEWAY : 129.34.0.0 : YORKTOWN-GATEWAY ::::
;
```

Step 2: Run MAKESITE

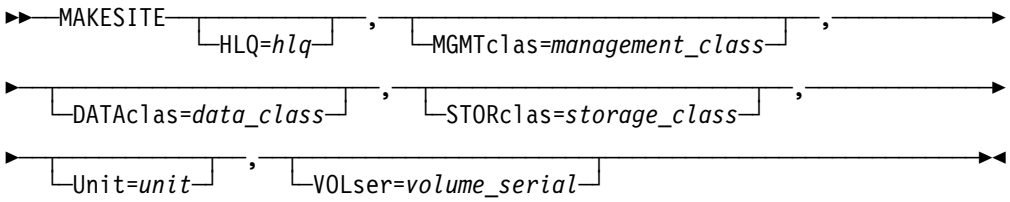
After you make changes to your *hlq*.HOSTS.LOCAL data set, you must generate and install new *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO data sets.

Because many servers and commands allocate *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO, it is important not to overwrite or delete these data sets while TCP/IP is running. To avoid disrupting any active users, use an HLQ that is different than your active HLQ. This will allow you to swap names (by renaming the old HOSTS data sets and then renaming the new HOSTS data sets) without starting and stopping TCP/IP.

MAKESITE Command

Use MAKESITE as a TSO command or in a batch job to generate new *hlq*.HOSTS.SITEINFO and *hlq*.HOSTS.ADDRINFO data sets. The parameters are the same for either a TSO command or a batch job invocation of MAKESITE.

Syntax



Parameters

HLQ=*hlq*

The high-level qualifier of both the input and output data sets. The name specified is appended to the HOSTS.LOCAL, HOSTS.SITEINFO and HOSTS.ADDRINFO data set names.

Minimum abbreviation: HLQ=,
Maximum length: 29 characters

MGMTCLAS=*management_class*

The SMS-managed management class. MGMTCLAS is valid only in an SMS environment.

Minimum abbreviation: MGMT=
Maximum length: 8 characters

DATACLAS=*data_class*

The SMS-managed data class. DATACLAS is valid only in an SMS environment.

MGMTCLAS=*management_class*
The SMS-managed management class. MGMTCLAS is valid only in an SMS environment.

Minimum abbreviation: MGMT=
Maximum length: 8 characters

DATACLAS=*data_class*
The SMS-managed data class. DATACLAS is valid only in an SMS environment.

Minimum abbreviation: DATA=
Maximum length: 8 characters

STORCLAS=*storage_class*
The SMS-managed storage class. STORCLAS is valid only in an SMS environment.

Minimum abbreviation: STOR=
Maximum length: 8 characters

UNIT=*unit*
An esoteric device name.

Minimum abbreviation: U=
Maximum length: 8 characters

VOLSER=*volume_serial*
Volume serial number.

Minimum abbreviation: VOL=
Maximum length: 6 characters

Usage Notes

- The optional parameters can be in any order
- Blanks are not allowed in the syntax
- MAKESITE gets its input from *hlq*.HOSTS.LOCAL, where the HLQ is derived in this order:
  - HLQ parameter specified either with the command or in the batch job
  - TSO userid or the TSO PROFILE PREFIX, if it is different from the userid. In a batch job, *userid* can come from any of several sources depending on the environment. It can be the user ID of the user who submitted the batch job, or it can be the batch job name.
  - The value specified with the DATASETPREFIX statement in TCPIP.DATA
  - System default

  The output data sets produced by MAKESITE are prefixed by either the HLQ parameter specified either on the command or batch job or the TSO userid or TSO PROFILE PREFIX, if it is different from the userid.
- Components that use the output from MAKESITE follow the standard naming conventions. If a DATASETPREFIX has been specified, it will be used as the high-level qualifier for HOSTS.SITEINFO and HOSTS.ADDRINFO.

Examples

If your current active HLQ was TCPIP.MVSA, you would follow these steps to run MAKESITE and rename the output data sets.

1. Run MAKESITE with the appropriate parameters to generate 2 new data sets from the new *hlq*.HOSTS.LOCAL data set.

   As a TSO command, you might enter:

   ```
   MAKESITE HLQ=TCPIP.H0004,MGMT=M0001,VOLSER=STRG01,UNIT=SYSDA
   ```

   As a batch job, you might use this JCL:

   ```
   //MAKESITE JOB ,TIME=2,NOTIFY=USER7
   //*
   //BATCH  EXEC PGM=MAKESITE,REGION=8000K,
   //  PARM='VOLSER=STRG01,UNIT=SYSDA,HLQ=TCPIP.H0004,MGMT=M0001'
   //*
   //STEPLIB DD DISP=SHR,DSN=TCPIP.V3R1.SEZALINK
   //SYSPRINT  DD  SYSOUT=*,DCB=(LRECL=132,RECFM=FBA,BLKSIZE=3960)
   //SYSABEND  DD  SYSOUT=*
   //
   ```

   Note the following:

   - This JCL is not shipped with TCP/IP
   - The size of the parameter string is limited to 100 bytes
   - Keywords in the parameter string can be abbreviated as shown in the MAKESITE syntax descriptions
   - Region size varies according to your configuration. Make sure that the region size specified is valid for your configuration.

   This will create TCPIP.H004.HOSTS.SITEINFO and TCPIP.H0004.HOSTS.ADDRINFO.

2. Rename your existing HOSTS.SITEINFO and HOSTS.ADDRINFO data sets. These data sets are currently accessed by TCP/IP users on the system and should not be deleted while TCP/IP is running.

   For example, change TCPIP.MVSA.HOSTS.SITEINFO to TCPIP.MVSA.HOSTS.SITEOLD and TCPIP.MVSA.HOSTS.ADDRINFO to TCPIP.MVSA.HOSTS.ADDROLD.

3. Rename the new HOSTS.ADDRINFO and HOSTS.SITEINFO data sets to replace the old ones.

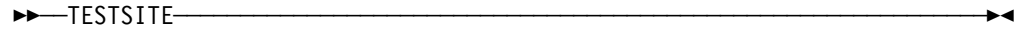
   For example, change TCPIP.H0004.HOSTS.SITEINFO to TCPIP.MVSA.HOSTS.SITEINFO and TCPIP.H0004.HOSTS.ADDRINFO to TCPIP.MVSA.HOSTS.ADDRINFO.

Testing the Site Table: After running MAKESITE, you can test the correctness of the *hlq*.HOSTS.ADDRINFO and *hlq*.HOSTS.SITEINFO data sets with TESTSITE.

TESTSITE Command

Use TESTSITE to verify that the *hlq*.HOSTS.ADDRINFO and *hlq*.HOSTS.SITEINFO data sets can correctly resolve the name of a host, gateway, or net.

Syntax



Parameters

Examples

To test your HOSTS data sets, enter:

```
TESTSITE
```

When prompted for a name, enter the host, gateway or net name you want to verify.

When you have checked all the names in question, enter `QUIT` and press `ENTER`.

Usage Notes

TESTSITE gets its input from the *hlq*.HOSTS.ADDRINFO and *hlq*.HOSTS.SITEINFO data sets, where the HLQ is derived in this order:

- TSO userid or the TSO PROFILE PREFIX, if it is different from the userid
- The value specified with the DATASETPREFIX statement in PROFILE.TCPIP and TCPIP.DATA
- System default

Part 2. Configuring the Servers

Chapter 6. Configuring the OE Telnet Server

This chapter contains information about installing, starting, stopping, and administering OE Telnet.

Installation Information

OE Telnet code is installed in the hierarchical file system (HFS) (path /usr/lpp/tcpip/sbin/otelnetd with a symbolic link to /usr/sbin/otelnetd) and in the MVS data set *hlq*.SEZALINK. The *hlq*.SEZALINK data set needs to be a PADS protected data set if you are running with the BPX.DAEMON facility class defined. OE checks whether the sticky bit is set on in the HFS. If it finds the sticky bit on, it first checks for an executable file in the MVS data set. If it does not find the executable file in a data set in the MVS search order, OE then uses the executable file in the HFS.

The sticky bit is on when otelnetd is distributed. If, after installation, you turn the sticky bit off, the executable file is loaded from the HFS (/usr/sbin/otelnetd) instead of being loaded from the MVS data set.

Notes: To run OE Telnet in a BPX.DAEMON facility class defined, the user must access the executable file in a PADs-protected data set.

OE telnet relies on the OE chcp command to provide code page conversions. Therefore, if the telnet client wants to use a code page other than the default IBM-1047 code page, it has to make use of the OE chcp command. For more information, see *OS/390 OpenEdition Planning*.

The HFS files used in the OE Telnet server and their locations in the HFS are as follows:

/etc/services — The ports for each application are defined here.

/etc/syslog.conf — The configuration parameters for usage of syslogd are defined in this file. otelnetd writes to local1.

/etc/inetd.conf — The configuration parameters for all applications started by inetd are defined in this file.

The BPX.DAEMON facility requires that the userid for otelnetd have read access to BPX.DAEMON and be a superuser.

/usr/sbin/otelnetd

This is a symbolic link to /usr/lpp/tcpip/sbin/oteletd, where the executable file for the OE telnet server is stored.

If BPX.DAEMON is specified, then the sticky bit must be set on, and otelnetd must reside in an authorized MVS data set. Also, with BPX.DAEMON facility class defined, the C-RTL (SCEERUN) data set must reside in an MVS authorized data set.

/etc/banner — If -h is not specified in the /etc/inetd.conf, then an additional banner is expected to be printed to the client's screen. This banner should be stored here.

/bin/fomtlinp — The executable file for utmp entry is stored here. This code will update the utmp entry as well as generate the child.

/etc/utmpx
: This file is updated by the call to fsumoclp. It contains a list of all the users who are logged in with their associated tty.

/dev/ptypXXXX and /dev/ttypXXXX
: UNIX does not know about devices. It understands only streams of data. Files found in /dev/ptypXXXX correlate with information being passed between TELNETD software and the master pty. Files found in /dev/ttypXXXX correlate with information being passed between the master pty and the slave pty.

  Note: These device drivers are not only used by TELNET. For example, for every OMVS that is started up, a /dev/ptypXXXX and /dev/ttypXXXX are allocated for usage. For information on allocating more of these files for more connections, see *OS/390 OpenEdition Planning*.

/usr/lib/terminfo
: The terminal information that is verified during the tgetent call is stored here. This file requires permissions of 644. For more information, see *OS/390 OpenEdition Planning*.

/usr/lib/nls/msg/C/tnmsgs.cat
: The message catalog used by the OE Telnet server is stored here.

  Where the server looks for the message catalog depends on the value of NLSPATH and LANG environment variables. If you want to store the message catalog elsewhere, you need to change the NLSPATH or the LANG environment variables. If the message catalog does not exist, the software will default to the messages hard-coded within the software. These messages duplicate the English message catalog that is shipped with the product.

/usr/man/C/cat1/telnetd.1
: This file contains the associated manual (man) pages for the OE Telnet server. It provides online help for the user.

Starting, Stopping, and Administration of OE Telnet

The Telnet server (Telnetd) operates by allocating a pty (pseudo-terminal device) for a client. It then creates a login process that has the slave side of the pty as stdin, stdout and stderr. Telnetd manipulates the master side of the pty, implementing the Telnet protocol and passing characters between the remote client and the login process.

The following standards are supported:

- RFC 854 Telnet Protocol Specification
- RFC 855 Telnet Option Specification
- RFC 856 Telnet Binary Transmission
- RFC 857 Telnet Echo Option
- RFC 858 Telnet Suppress Go Ahead Option
- RFC 859 Telnet Status Option
- RFC 860 Telnet Timing Mark Option
- RFC 861 Telnet Extended Options - List Option
- RFC 885 Telnet End of Record Option
- RFC 1073 Telnet Window Size Option

- RFC 1079 Telnet Terminal Speed Option
- RFC 1091 Telnet Terminal type option
- RFC 1096 Telnet X Display Location Option
- RFC 1123 Requirements for Internet Hosts -- Application and Support
- RFC 1184 Telnet Linemode Option
- RFC 1372 Telnet Remote Flow Control Option
- RFC 1571 Telnet Environment Option Interoperability Issues
- RFC 1572 Telnet Environment Option

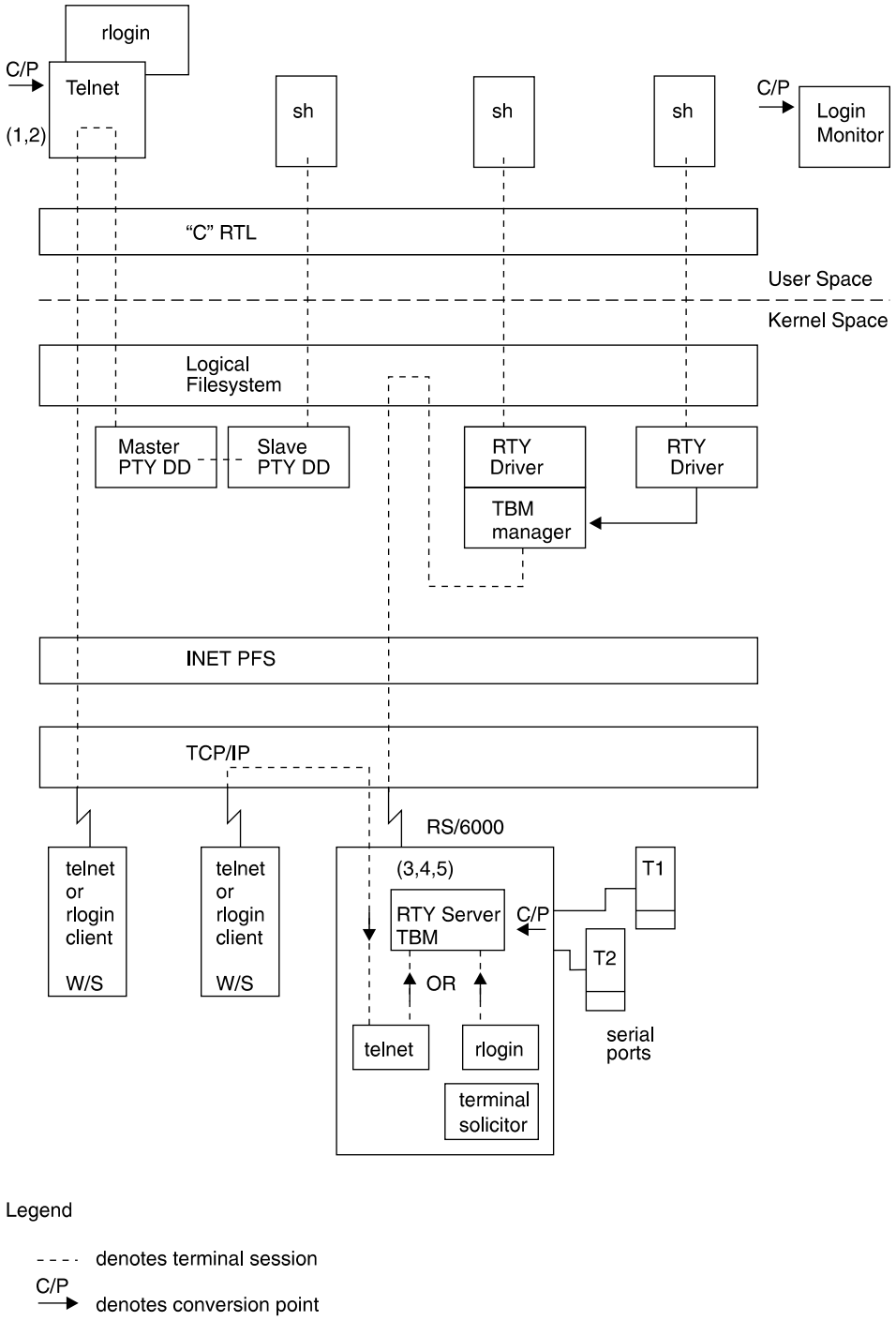


Figure 2. OpenEdition MVS Terminal Attachment Paths.. The following terminal attachment paths are illustrated:

- *Paths 1 and 2 are Telnet and rlogin paths from remote clients to servers located on the Open Edition host system.*
- *Paths 3, 4 and 5 are attachment paths provided by OCS and Telnet and rlogin servers are outbound on the RISC System/6000. Serial terminal attachment is also provided by the RISC System/6000.*

When an OE Telnet session is started up, otelnetd sends Telnet options to the client side indicating a willingness to do the following:

- DO TERMINAL TYPE
- DO TSPEED
- DO XDISPLOC
- DO NEW-ENVIRON
- DO ENVIRON
- WILL SUPPRESS GO AHEAD
- DO ECHO
- DO LINEMODE
- DO NAWS
- WILL STATUS
- DO LFLOW
- DO TIMING-MARK

With the following OE Telnet options, OE telnetd has support for enabling **LOCALLY.**

- WILL BINARY

  This option indicates that the client is willing to send 8 bits of data, rather than the normal 7 bits of network virtual terminal data.

- WILL ECHO

  When the LINEMODE option is enabled, a WILL ECHO or WONT ECHO will be sent to the client to indicate the current state of terminal echoing. When terminal echo is not desired, a WILL ECHO is sent to indicate that OE telnetd will take care of echoing any data that needs to be echoed to the terminal, and then nothing is echoed. When terminal echo is desired, a WONT ECHO is sent to indicate that OE telnetd will not be doing any terminal echoing, so the client should do any terminal echoing that is needed.

- WILL LOGOUT

  When a DO LOGOUT is received, a WILL LOGOUT is sent in response and the Telnet session is shut down.

- WILL SGA

  This option indicates that it will not be sending IAC GA, the go ahead command.

- WILL STATUS

  Indicates a willingness to send the client, upon request, the current status of all Telnet options.

- WILL TIMING-MARK

  Whenever a DO TIMING-MARK is received, a WILL TIMING-MARK is the response. It is only used in kludge linemode support.

With the following OE Telnet options, OE Telnetd has support for enabling **REMOTELY.**

- DO BINARY

  Sent to indicate that OE Telnetd is willing to receive an 8-bit data stream.

- DO ECHO

If a WILL ECHO is received, a DONT ECHO will be sent in response.

- DO ENVIRON

  Indicates a desire to be able to request environment variable information. (See RFC 1408.)

- DO LFLOW

  Requests that the client handle flow control characters remotely.

- DO LINEMODE

  Supports requests that the client do line-by-line processing.

- DO NAWS

  Requests that the client inform the server when the window size changes.

- DO NEW-ENVIRON

  Indicates a desire to be able to request environment variable information. (See RFC 1572.)

- DO SGA

  Indicates that it does not need to receive IAC GA, the go ahead command.

- DO TERMINAL-TYPE

  Indicates a desire to be able to request the name of the type of terminal that is attached to the client side of the connection.

- DO TERMINAL-SPEED

  Indicates a desire to be able to request information about the speed of the serial line to which the client is attached.

- DO TIMING-MARK

  Only supported if the client responded with WONT LINEMODE. If the client responds with WILL TM, then it is assumed that the client will support kludge linemode. It is not used for any other purposes.

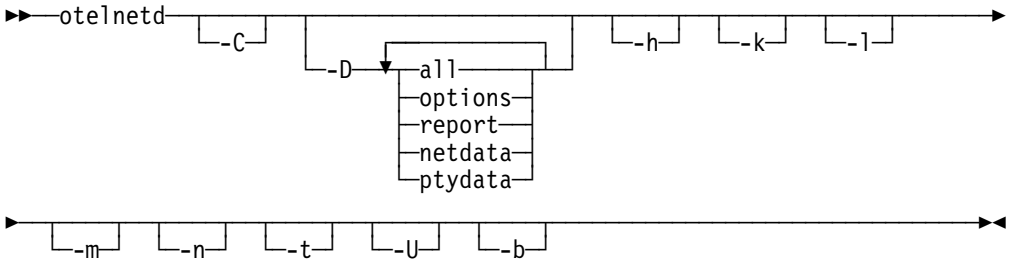
- DO XDISPLOC

  Indicates a desire to be able to request the name of the X Window System display that is associated with the Telnet client.

otelnetd

The following syntax is used in the /etc/inetd.conf file to define the arguments used to invoke otelnetd.

Syntax



Parameters

-C

Prints user messages in uppercase. There are several exceptions. Messages issued on startup are not affected by the -C option because the -C option is not processed during the startup. Also, data transmittal messages will not be uppercase. Data transmittal messages are the output from the -D netdata option or the -D ptydata option.

-D

The -D option has several suboptions:

options Prints information about the negotiation of Telnet options. This is used for debugging purposes. It allows telnetd to print out debugging information to the connection, allowing the user to see what telnetd is doing.

report Prints the options information, plus some additional information about what processing is going on. This also includes print information slated for suboption=options. This is used for debugging purposes. It allows telnetd to print out debugging information to the connection, to enable the user to see what telnetd is doing.

netdata Displays the data stream received by telnetd. This is used for debugging purposes. It allows telnetd to print out debugging information to the connection, to enable the user to see what telnetd is doing.

ptydata Displays the data stream written to the pty. This is used for debugging purposes. It allows telnetd to print out debugging information to the connection, to enable the user to see what telnetd is doing.

all Supports all options: options, report, netdata, and ptydata.

-h

Disables the display of the /etc/banner file at the user's terminal.

Internal processing to support this option is simple. A variable hostinfo is initialized to one. If this option is specified, during processing of this option, hostinfo is reset to zero. This variable is then used to determine if the routine that issues the banner should be called.

-k

This option specifically states DON'T INITIATE KLUDGE LINEMODE.

This option is useful only if telnetd has been compiled with both linemode and kludge linemode support. If the k option is NOT specified, AND if the remote client does not support the LINEMODE option, then we will try Kludge Linemode, telnetd will operate in character at a time mode. It will still support kludge linemode, but will only go into kludge linemode if the remote client

requests it. (This is done by the client sending DONT SUPPRESS-GO-AHEAD and DONT ECHO.) The k option is most useful when there are remote clients that do not support kludge linemode, but pass the heuristic (if they respond with WILL TIMING-MARK in response to a DO TIMING-MARK) for kludge linemode support.

-l

Specifies line mode; tries to force clients to use line mode. If the LINEMODE option is not supported, it will go into kludge linemode.

-m

Enables creation of a forked or spawned process to coexist in the same address space. Using this option improves performance.

-n

Disable TCP keep-alives. Normally telnetd enables the TCP keep-alive mechanism to probe connections that have been idle for some period of time to determine if the client is still there, so that idle connections from machines that have crashed or can no longer be reached can be cleaned up. The cleanup of disabled connections is controlled by the presence of the KEEPALIVEOPTIONS statement in the TCPIP profile.

-t

Internal tracing, intended to replace the DIAGNOSTICS compile option currently in place within the BSD code. It will also turn on the REPORT option, as if the user also specified -D Report.

-U

This option causes telnetd to refuse connections from any address that cannot be mapped back into a symbolic name via the gethostbyaddr(3) routine.

Internal processing to support this option is simple. A variable registerd_host_only is initialized to zero. If this option is specified, during processing of this option, registerd_host_only is reset to one. This variable is then used to determine if the output of the gethostbyaddr function should be accepted if it returns a NULL value.

-b

This option forces the server to DO BINARY in the first pass during negotiations with the client.

SMF Record Handling

The SMF records generated are the typical set of records that MVS generates for start of job (login) and end of job (logoff). Additionally, interval records can be issued during the life of the user login. These records are SMF TYPE 30 and TYPE 72 and not the TYPE 118 in the current Telnet server. The process of issuing these records is external to the specific daemons.

Chapter 7. Configuring the OE File Transfer Protocol (FTP) Server

Before You Configure...:

Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

This chapter describes how to configure the File Transfer Protocol (FTP) server. It also provides information about using the MODIFY command to dynamically control tracing. Information about security considerations for the FTP server is described in “Security Considerations for the FTP Server” on page 154.

Although the MVS FTP client cannot issue the requests, this server supports the following requests from other FTP clients:

PASV	Permits transfer of files to and from a third party server.
STRU R	Transfers data in record structure instead of file structure.

Configuration Process

Steps to configure the FTP server:

1. Specify port and KEEPALIVEOPTIONS information
2. Update /etc/services with the port to be reserved for the FTP server
3. Update the FTPD cataloged procedure *hlq*.SEZAINST(FTPD). FTPD is also known by the SMP/E distribution name EZAFTPAP.
4. Specify FTP configuration statements in FTP.DATA
5. Configure the FTP server for SMF (optional)
6. Configure the FTP user-written exits (optional)
7. Specify configuration statements in TCPIP.DATA
8. Install the SQL Query Function (optional)
9. Update /etc/syslog.conf for the FTP server.

To dynamically enable or disable the trace options, see “Starting, Stopping, and Tracing the OE FTP Server” on page 188.

Step 1: Specify Port and KEEPALIVEOPTIONS Information

To ensure that ports are reserved for the FTP server, you must reserve them for OMVS in the PROFILE.TCPIP data set and specify their precise use in the /etc/services file. For example, to reserve port 20 for data transfers and port 21 for incoming control connection requests, add the following to the PROFILE.TCPIP data set:

```
PORT
  21 TCP OMVS        ; FTP server control port
  20 TCP OMVS        ; FTP server data port
```

To allow the FTP data connections to timeout when there has been no activity on the data connection for a certain amount of time, add the KEEPALIVEOPTIONS statement to the PROFILE.TCPIP data set:

```
KEEPALIVEOPTIONS INTERVAL number_of_minutes ENDKEEPALIVEOPTIONS
```

Be careful when choosing a timeout interval for the KEEPALIVEOPTIONS statement because this value will affect *all TCP* connections at this host for which KEEPALIVEOPTIONS has been activated, not just the FTP data connections.

See “PORT Statement” on page 83 for more information on the PORT statement.
See “KEEPALIVEOPTIONS Statement” on page 78 for more information on the KEEPALIVEOPTIONS statement.

Step 2: Update /etc/services

In /etc.services, add:

```
ftp 21/tcp
```

Note:

In the /etc/services file, only one port (the one for the control connection) is listed.

Step 3: Update the FTPD Cataloged Procedure

Update the FTP cataloged procedure FTPD by copying the sample in *hlq*.SEZAINST(FTPD) to your system or recognized PROCLIB and modifying it to suit your local configuration. Specify FTPD parameters and change the data set names as required.

The FTP cataloged procedure is also known by the SMP/E distribution name EZAFTPAP.

FTP Server Cataloged Procedure (FTPD)

```
//FTPD   PROC MODULE='FTPD',PARMS=''
//*********************************************************************
//*                                                                   *
//*        TCP/IP for MVS                                             *
//*                                                                   *
//*        Descriptive Name:          FTP Server Start Procedure      *
//*                                                                   *
//*        File Name:                 tcpip.SEZAINST(EZAFTPAP)        *
//*                                   tcpip.SEZAINST(FTPD)            *
//*                                                                   *
//*        SMP/E Distribution Name:   EZAFTPAP                        *
//*                                                                   *
//*                                                                   *
//*           Licensed Materials - Property of IBM                    *
//*           This product contains "Restricted Materials of IBM"     *
//*           5645-001 5655-HAL (C) Copyright IBM Corp. 1995, 1997.   *
//*           All rights reserved.                                    *
//*           US Government Users Restricted Rights -                 *
//*           Use, duplication or disclosure restricted by            *
```

```
//*            GSA ADP Schedule Contract with IBM Corp.            *
//*            See IBM Copyright Instructions.                     *
//*                                                                *
//*                                                                *
//******************************************************************
//FTPD   EXEC PGM=&MODULE,REGION=4096K,TIME=NOLIMIT,
//      PARM='POSIX(ON) ALL31(ON)/&PARMS'
//CEEDUMP  DD SYSOUT=*
//*
//*       SYSFTPD is used to specify the FTP.DATA file for the FTP
//*       server.  The file can be any sequential data set, member
//*       of a partitioned data set (PDS), or HFS file.
//*
//*       The SYSFTPD DD statement is optional.  The search order for
//*       FTP.DATA is:
//*
//*          /etc/ftp.data
//*          SYSFTPD DD statement
//*          jobname.FTP.DATA
//*          SYS1.TCPPARMS(FTPDATA)
//*          tcpip.FTP.DATA
//*
//*       If no FTP.DATA file is found, FTP default values are used.
//*       For information on FTP defaults, see the Customization
//*       and Administration Guide and TCP/IP OE MVS Applications
//*       Feature Guide.
//*SYSFTPD DD DISP=SHR,DSN=TCPIP.SEZAINST(FTPSDATA)
//*
//*      SYSTCPD explicitly identifies which file is to be
//*      used to obtain the parameters defined by TCPIP.DATA.
//*      The SYSTCPD DD statement should be placed in the JCL of
//*      the server.  The file can be any sequential data set,
//*      member of a partitioned data set (PDS), or HFS file.
//*SYSTCPD DD DISP=SHR,DSN=TCPIP.SEZAINST(TCPDATA)
//*
//*      SYSFTSX explicitly identifies which file is to be used
//*      for the EBCDIC-ASCII translation table.  The file can
//*      be any sequential data set, member of a partitioned data
//*      set (PDS), or HFS file.
//*SYSFTSX DD DISP=SHR,DSN=TCPIP.STANDARD.TCPXLBIN
```

The REGION size requirement for the FTPD address space might increase under certain circumstances. REGION=7500K is provided as a minimum requirement.

Specifying the FTPD Parameters

The system parameters required by the FTP server are passed by the PARM parameter on the EXEC statement of the FTPD cataloged procedure. Add your parameters to `PARMS='` in the PROC statement of the FTPD cataloged procedure, making certain that:

- Each parameter is separated by a blank
- All parameters are in uppercase

For example: `//FTPD PROC MODULE='FTPD',PARMS='TRACE ANONYMOUS PORT 621'`

ANONYMOUS
Allows remote users to enter ANONYMOUS as a user ID and log on without supplying a logon password. Specifying ANONYMOUS makes your universally permitted data sets accessible to all users on the TCP/IP network.

ANONYMOUS=*user_id*
Allows a remote user to enter ANONYMOUS as a user ID. When ANONYMOUS is entered as the user ID, the FTP server treats the login request as though the specified *user_id* was entered instead of ANONYMOUS. The user will be prompted for the password to *user_id* and, if the user enters the correct password, the user will be logged in as the specified *user_ID*.

ANONYMOUS=*user_id/password*
Allows a remote user to enter ANONYMOUS as a user ID. When ANONYMOUS is entered as the user ID, the FTP server treats the login request as though the specified *user_id* was entered instead of ANONYMOUS. The FTP server automatically provides the *password* for the specified *user_id* and the user will be logged in as the specified *user_ID*.

AUTOMOUNT
Permits a DASD volume to be mounted when attempts are made to access data sets on that volume.

AUTORECALL
Permits data sets migrated by a storage manager, such as Hierarchical Storage Manager (HSM), to be recalled automatically.

DATASETMODE
Treats all lower qualifiers of address space names as part of the same directory. This affects the behavior of DIR, LS, MGET, and MDLETE because all lower qualifiers are returned.

DIRECTORYMODE
Treats each level of an address space name as if it were a directory. This affects the behavior of DIR, LS, MGET, and MDLETE because only the next lower qualifier is returned.

INACTIVE *number_seconds*
Sets the inactivity time-out to the specified number of seconds. A control connection that is inactive for this amount of time is closed. The default inactivity time-out is 300 seconds (5 minutes). The maximum inactive time is 86 400 seconds. A value of 0 will disable the inactivity timer and inactive control connections will not time out.

NOAUTOMOUNT
Prevents a DASD volume from being mounted when attempts are made to access data sets on that volume.

NOAUTORECALL
Prevents data sets migrated by a storage manager, such as HSM, from being recalled automatically. Migrated data sets can still be deleted even though NOAUTORECALL is specified.

Note: Only sequential and whole partitioned data sets can be deleted without recalling. Partitioned data set members require the whole data set to be recalled.

PORT *port_num*
Accepts incoming requests on the specified (decimal) port number rather than the port specified in /etc/services or the default port of 21. (*port_num* – 1) will be used for data transfer. The maximum port number is 65534.

TRACE
Displays tracing information to syslog. Running TRACE will affect performance. TRACE should be specified only if you need information for the IBM TCP/IP support group.

Step 4: Specify FTP Configuration Statements in FTP.DATA

The FTP.DATA data set is optional. The FTP daemon looks for this data set during initialization, following this sequence:

1. /etc/ftp.data
2. A data set specified by the //SYSFTPD DD statement
3. *ftpserve_job_name*.FTP.DATA
4. SYS1.TCPPARMS(FTPDATA)
5. *hlq*.FTP.DATA data set

If you use an MVS data set, this data set should have a logical record length of 80 and a block size that is a multiple of 80.

The default values for the FTP server parameters are in the FTPD module. You can change these defaults using statements in the FTP.DATA configuration data set.

It is not necessary to include all statements in the FTP.DATA data set. Only include the statements if the default value is not what you want, since the default will be used for any statement not included in the FTP.DATA data set.

Several of the FTP server parameters can be changed during an FTP session by issuing the SITE subcommand from the FTP client. See *TCP/IP for MVS: User's Guide* for more information.

The FTP.DATA data set can also be used to change the defaults for the FTP client local site parameters. See *TCP/IP for MVS: User's Guide* for more information about using the FTP.DATA data set for the FTP client local site parameters.

Summary of FTP Server Configuration Statements

The statements for the FTP.DATA data set are summarized in Table 8 on page 138 and explained in detail in “FTP.DATA Data Set Statements” on page 154.

If you plan to share the FTP server FTP.DATA data set with the FTP client, note that some of the values for the statements in the FTP.DATA data set have different meanings in the two environments. If the files are shared, error messages might be generated or values that are not valid might be used for each client that uses the FTP.DATA data set containing server-only keywords. To avoid these errors, use

separate FTP.DATA data sets for the FTP client and the FTP server if you are specifying any conflicting keywords.

Table 8 (Page 1 of 2). Summary of FTP Server Configuration Statements

Statement	Description	Page
ANONYMOUS	Allow a remote user to issue USER ANONYMOUS without supplying a logon password	155
ASATRANS	Specify how print control characters should be handled	156
AUTOMOUNT	Specify whether to mount DASD volumes containing data sets to be accessed	156
AUTORECALL	Automatically recall data sets migrated by the storage manager	157
AUTOTAPEMOUNT	Specify whether to mount tape volumes containing data sets to be accessed	157
BLKSIZE	Specify the block size of newly allocated data sets	158
BUFNO	Specify the number of access method buffers	159
CHKPTINT	Specify the checkpoint interval when the FTP server is the sending site in a file transfer request	159
CONDDISP	Keep and catalog or delete a data set when a file transfer ends prematurely	160
CTRLCONN	Specify ASCII codeset to be used for the control connection	160
DATACLASS	Specify the SMS-managed data class as defined by your organization for the FTP server	161
DB2	Specify the name of the DB2 subsystem	162
DB2PLAN	Specify the name of the DB2 plan to be used by the FTP server	163
DCBDSN	Specify a data set to be used as a model for allocation of new data sets	163
DEST	Specify the NJE destination to which the files are routed when you enter a PUT command	164
DIRECTORY	Specify the number of directory blocks to be allocated for the directory of a PDS	165
DIRECTORYMODE	Specify how to treat the data set qualifiers below the current directory	166
FILETYPE	Specify the operational mode of the server	166
INACTIVE	Set the inactivity timer to a specified number of seconds	167
JESLRECL	Specify the record length of the job being submitted	167
JESPUTGETO	Specify the number of seconds of the JES PutGet time-out	168
JESRECFM	Specify the record format of the job being submitted	168
LRECL	Specify the size of the records in a data set	169
MGMTCLASS	Specify the SMS management class to be assigned to newly allocated data sets	170
MIGRATEVOL	Specify the volume ID for migrated data sets not under the control of IBM storage management systems	170
PRIMARY	Specify the amount of tracks, blocks, or cylinders for primary allocation	171
QUOTESOVERRIDE	Specify use of single quotes in filename	172

Table 8 (Page 2 of 2). Summary of FTP Server Configuration Statements

Statement	Description	Page
RDW		172
RECFM	Specify the record format of a data set	172
RETPD	Specify the number of days that a newly allocated data set should be retained	174
SBDATACONN	Specify single-byte ASCII/EBCDIC conversion for the data connection	175
SECONDARY	Specify the amount of tracks, blocks, or cylinders for secondary allocation	176
SMF	Specify the default SMF record subtype for all SMF records	176
SMFAPPE	Specify the SMF record subtype for the APPEND subcommand	177
SMFDEL	Specify the SMF record subtype for the DELETE subcommand	178
SMFEXIT	Call the FTPSMFEX user exit routine	179
SMFJES	Collect SMF records when FILETYPE is JES	179
SMFLOGN	Specify the SMF record subtype when recording logon failures	179
SMFREN	Specify the SMF record subtype for the RENAME subcommand	180
SMFRETR	Specify the SMF record subtype for the RETR subcommand	181
SMFSQL	Collect SMF records when FILETYPE is SQL	181
SMFSTOR	Specify the SMF record subtype for the STOR and STOU subcommands	182
SPACETYPE	Specify whether newly allocated data sets are allocated in blocks, cylinders, or tracks	182
SPREAD	Specify output in spreadsheet format when file type is SQL	183
SQLCOL	Specify the column headings of the output file	183
STORCLASS	Specify the SMS-managed storage class for the FTP server	184
TRACE	Start tracing for the FTP server	185
TRAILINGBLANKS	Include trailing blanks in fixed format data sets when retrieved	185
UMASK	Specify the file mode creation mask.	185
UNITNAME	Specify the unit type for allocation of new data sets	186
VOLUME	Specify the volume serial number for allocation of new data sets	187
WRAPRECORD	Specify whether data will be wrapped or truncated if no new line character is encountered before the logical record length is reached	188

Sample FTP Server Configuration Data Set (FTPSDATA)

```
;*******************************************************************
;
;   Name of File:              tcpip.SEZAINST(FTPSDATA)
;
;   Descriptive Name:          FTP.DATA  (for OE-FTP Server)
;
;   SMP/E Distribution Name:  EZAFTPAS
```

```
;
;         5645-001 5655-HAL (C) Copyright IBM Corp. 1997.
;         Licensed Materials - Property of IBM
;         This product contains "Restricted Materials of IBM"
;         All rights reserved.
;         US Government Users Restricted Rights -
;         Use, duplication or disclosure restricted by
;         GSA ADP Schedule Contract with IBM Corp.
;         See IBM Copyright Instructions.
;
;   This FTP.DATA file is used to specify default file and disk
;   parameters used by the FTP server.
;
;   Syntax Rules for the FTP.DATA Configuration File:
;
;   (a) All characters to the right of and including a ; will be
;       treated as a comment.
;
;   (b) Blanks and <end-of-line> are used to delimit tokens.
;
;   (c) The format for each statement is:
;
;       parameter value
;
;*********************************************************************
;
;ANONYMOUS                      ; anonymous login accepted
;ASATRANS      FALSE            ; do NOT translate control characters
                                     ; in ASA text
AUTOMOUNT     TRUE             ; automatic mount of unmounted volume
AUTORECALL    TRUE             ; automatic recall of migrated data sets
;AUTOTAPEMOUNT FALSE    ; do NOT automatically mount tape volumes
BLOCKSIZE     6233              ; new data set allocation blocksize
CONDDISP      CATLG             ; data sets catalogued if transfer fails
;CTRLCONN      IBM-850          ; ascii code set for control connection
;DATACLASS     SMSDATA     ; sms data class name
;DB2           D31                ; db2 subsystem name
;DB2PLAN       PLANNAME     ; db2 plan name for OE-FTP
;DCBDSN        MODEL.DCB     ; new data set allocation model dcb name
;DEST          USER14@MVSL  ; files destination for store
DIRECTORY     27                ; new data set allocation directory blocks
DIRECTORYMODE FALSE      ; directorymode vs. data set mode
FILETYPE      SEQ               ; file transfer mode
INACTIVE      300               ; inactive time out
JESLRECL      80                ; lrecl of jes jobs
JESPUTGETTO   600               ; timeout for remote job submission put/ge
JESRECFM      F                 ; recfm of jes jobs
LRECL         256               ; new data set allocation lrecl
;MGMTCLASS     SMSMGMT   ; sms mgmtclass name
;MIGRATEVOL    MIGRAT     ; migration volume volser
PRIMARY       1                 ; new data set allocation primary space
;QUOTESOVERRIDE FALSE    ; single quote(s) are treated as part of
                                     ; hfs filename, i.e. single quotes do
```

```
                                    ; NOT indicate working directory override
RECFM         VB                   ; new data set allocation record format
;RETPD                             ; no data set allocation retention period
;RETPD        30                   ; new data set retention period: 30 days
;RETPD        0                    ; retention period will expire same day
                                    ; the file is created
;SBDATACONN   (IBM-1047,IBM-850) ; ebcdic/ascii code sets for data conn.
SECONDARY     1                    ; new data set allocation secondary space
;SMF          76                   ; SMF record subtype for all SMF records
;SMFAPPE      70                   ; SMF record subtype for APPE records
;SMFDEL       71                   ; SMF record subtype for DELE records
;SMFEXIT                           ; load SMF user exit FTPSMFEX
;SMFJES                            ; SMF recording when filetype=jes
;SMFLOGN      72                   ; SMF record subtype for LOGN records
;SMFREN       73                   ; SMF record subtype for REN records
;SMFRETR      74                   ; SMF record subtype for RETR records
;SMFSQL                            ; SMF recording when filetype=sql
;SMFSTOR      75                   ; SMF record subtype for STOR/STOU record
SPACETYPE     TRACK                ; new data set allocation space type
SPREAD        FALSE                ; sql output format
SQLCOL        NAMES                ; sql output uses column names as headings
;STORCLASS    SMSSTOR              ; sms storclass name
;TRACE                             ; trace active
;TRAILINGBLANKS TRUE               ; include trailing blanks when fixed
                                    ; format data sets are retrieved
;UMASK        027                  ; octal UMASK to restrict setting
                                    ; of permission bits when creating
                                    ; new hfs files
;UNITNAME     3380                 ; new data set allocation unit
;VOLUME       WRKLB2               ; new data set allocation volume serial
WRAPRECORD    FALSE                ; data is NOT wrapped to next record
```

Specifying Attributes for New MVS Data Sets

When allocating new data sets, there are two methods you can use to specify the data set attributes. You can individually use the data set attribute parameters with the SITE command or the statements in the FTP.DATA data set. Or, if your system programmer has used the Storage Management System to group together default attributes into named classes, you can specify those class names on the DATACLASS, STORCLASS, and MGMTCLASS statements.

Dynamic Allocation: The FTP server allows a client program to dynamically allocate a new physical sequential data set or a partitioned data set (PDS) for the purpose of transferring data to be written to that data set. The following optional allocation variables can be used to override and turn off the hard-coded defaults that affect the allocation of the data set.

Variable	**FTP.DATA statement**
allocation units	SPACETYPE
blocksize	BLKSIZE
data class	DATACLASS
directory blocks	DIRECTORY
logical record length	LRECL
management class	MGMTCLASS

model DCB values	DCBDSN
primary space	PRIMARY
secondary space	SECONDARY
record format	RECFM
retention period	RETPD
storage class	STORCLASS
unit	UNITNAME
volume serial number	VOLUME

Some of these allocation variables might provide duplicate information. For example, the model DCB might have a record format (RECFM) that differs from the record format specified by a data class and from the one explicitly specified by the client. FTP passes all variables that are specified to dynamic allocation and lets it determine which of the specifications take precedence. The following list describes the exceptions to that policy:

- If neither the primary nor secondary space quantity is specified, then the allocation units value is not sent.
- If the data set organization is physical sequential, then directory blocks specification is not sent.
- Otherwise, all variables are sent to dynamic allocation where the order of precedence is:
  1. Any FTP.DATA statements or SITE parameters explicitly specified or defaulted
  2. Any attributes picked up from the model DCB and not otherwise explicitly specified
  3. Any attributes picked up from the data class and not previously derived from 1 or 2
  4. Any allocation defaults

Storage Management Subsystem (SMS): You can specify one or more of the following SMS classes to manage characteristics that are associated with or assigned to data sets.

- Data class is an SMS construct that determines data set allocation attributes used by SMS for creation of data sets. The fields listed are available attributes that serve as a template for allocation. Each is ***optional*** and is overridden by any explicit specification of FTP allocation variables or by a model DCB (DCBDSN).

Variable	**FTP.DATA statement**
directory blocks	DIRECTORY
logical record length	LRECL
primary space	PRIMARY
record format	RECFM
retention period	RETPD
secondary space	SECONDARY

  Note: If either primary or secondary space is explicitly specified, then the primary and secondary values from data class are not used.

- Management class is an SMS construct that determines DFHSM action for data set retention, migration, backup, and release of allocated but unused space. Management class replaces and expands attributes that otherwise would be

specified. That is, management class might override any other specification of retention period.

- Storage class is a list of storage performance and availability services requests for an SMS-managed data set that SMS attempts to honor when selecting a volume or volumes for the data set. It might conflict with an explicit specification of volume and unit. If storage class is used, then volume and unit should be unspecified.

Step 5: Configure the FTP Server for SMF

The FTP server can write type 118 (X'76') SMF records to record transactions made by the FTP server. SMF records are independent of the IP connection. They are created for both Offload host connections and regular host connections. SMF records can be written for the following requests:

- APPEND
- DELETE
- RENAME
- RETRIEVE
- STORE
- STORE UNIQUE

Information about the previous requests can be recorded for:

- FTP server running in normal data transfer mode (FILETYPE=SEQ)
- FTP server running remote job submission (FILETYPE=JES)
- FTP server running Structured Query Language (SQL) queries (FILETYPE=SQL)
- Any combination of SEQ, JES, and SQL

For requests involving data transfer (APPEND, GET, PUT, RETR, or STOR) an SMF record will be written for both successfully and unsuccessfully completed data transfer requests which have begun data transfer. For data transfer requests which have completed unsuccessfully, the byte count of transmission field (offset 68) will contain the number of bytes transferred before the failure, and the recent server reply field (offset 73) will contain the 3-digit error reply code sent to the client.

The FTP server can also write SMF records when a logon attempt fails.

The capability also exists for a user-written exit routine to get control before the SMF records are written.

The following sections describe how to configure the FTP server for use with SMF.

Summary of FTP Server SMF Statements

If you want the FTP server to write type 118 (X'76') SMF records, you must include at least one of the SMF subtype statements (SMF, SMFAPPE, SMFDEL, SMFLOGN, SMFREN, SMFRETR, or SMFSTOR) in the FTP.DATA data set.

Table 9 on page 144 shows the SMF statements in FTP.DATA.

Table 9. Summary of FTP Server SMF Statements

Statement	Description	Page
SMF	Specify the default SMF record subtype for all SMF records	176
SMFAPPE	Specify the SMF record subtype for the APPE (APPEND) subcommand	177
SMFDEL	Specify the SMF record subtype for the DELE (DELETE) subcommand	178
SMFEXIT	Call the FTPSMFEX user exit routine	179
SMFJES	Collect SMF records when FILETYPE is JES	179
SMFLOGN	Specify the SMF record subtype when recording logon failures	179
SMFREN	Specify the SMF record subtype for the RNFR / RNTO (RENAME) subcommand	180
SMFRETR	Specify the SMF record subtype for the RETR (RETRIEVE) subcommand	181
SMFSQL	Collect SMF records when FILETYPE is SQL	181
SMFSTOR	Specify the SMF record subtype for the STOR (STORE) and STOU (STORE UNIQUE) subcommands	182

If SMF subtype statements are not coded in the FTP.DATA data set, then no SMF records are written by the FTP server.

FTP Server SMF User Exit

The FTP server SMF user exit is called before an SMF record that contains information about an FTP server session is written to the SYS1.MANx data set. The user exit allows site specific modifications to the record and controls whether the record is written to the SYS1.MANx data set.

Specify the FTP server SMF user exit option by including the SMFEXIT statement in the FTP.DATA data set. If this option is not specified, the system writes all FTP server SMF records specified in the FTP.DATA data set to the SYS1.MANx data set.

You must name this user exit routine FTPSMFEX, and place it in an installation-defined link library or an APF-authorized data set defined by a STEPLIB DD statement in the FTPD cataloged procedure. FTP calls the SMF user exit before each SMF record is written.

On entry to FTPSMFEX, register 1 contains a pointer to the following 2-word parameter list.

Offset Value

0 Pointer to the return code

4 Pointer to the SMF record

Prior to calling the SMF user exit, the return code is set to zero. A zero return code specifies that the SMF record will be written. To suppress writing of the SMF record to the SYS1.MANx data set, the user exit must change the return code to a non-zero value.

Appendix A, “SMF Records” on page 249 contains descriptions of TCP/IP SMF records.

Example FTPSMFEX User Exit

The following example shows an FTPSMFEX user exit.

```
*********************************************************************
* Function:   Allow FTP Server SMF record recording only when       *
*             the client is outside subnet 9.24.104                 *
*********************************************************************
*
FTPSMFEX CSECT
FTPSMFEX AMODE 31
FTPSMFEX RMODE ANY
         SAVE  (14,12)
         BALR  12,0
         USING *,12
         B     BEGIN
         DC    C'FTPSMFEX '
         DC    C' &SYSDATE '
         DC    C' &SYSTIME '
         DS    0F
BEGIN    LR    R2,R1                   *Parm pointer
         USING PARMS,R2
         L     R4,PTRRC                *-> Return code field
         L     R9,PTRSMFR              *-> SMF record
         USING SMFREC,R9
         L     R3,SMFREMIP             *Foreign IP Address
         SRL   R3,8                    *Get rid of the 8 loworders
         SLL   R3,8                    *Back into line again
         LM    R5,R7,OURBXLE           *Adresses for net loop
NETLOOP  C     R3,0(R5)                *One of our subnets ?
         BE    SKIPREC                 *- Yes, Do not write SMF record
         BXLE  R5,R6,NETLOOP           *Loop through all subnets
         SR    R15,R15                 *Not one of ours - write SMFrec
         B     DONE
SKIPREC  LA    R15,4                   *Do not write record
DONE     ST    R15,0(R4)               *Return the RC
         RETURN (14,12),RC=(15)    RETURN TO CALLER
OURBXLE  DC    A(OURSUB,4,OURSUBSL-4)
OURSUB   DC    AL1(9,24,104,0)                *9.24.104.0 (Production)
OURSUBSL EQU   *
R0       EQU   0
R1       EQU   1
R2       EQU   2
```

```
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
*
PARMS    DSECT
PTRRC    DC    F'0'
PTRSMFR  DC    F'0'
*
SMFREC   DSECT                        *FTP Server SMF record
         DC    24X'00'                *Std. SMF header
SMFCMD   DC    CL4' '                 *FTP subcommand
SMFFTYPE DC    CL4' '                 *File type (SEQ,JCL,SQL)
SMFREMIP DC    AL4(0)                 *Foreign host IP address
SMFLOCIP DC    AL4(0)                 *Local IP address
         DS    0F                     *Remainder of record not used
*
         END
```

Step 6: Configure the User Written Exits

To limit access to an FTP server, you can use any of the user exits described in this section. The FTP server provides increased security by using 4 user exits. The user exit load modules can be placed in any APF-authorized library for which the FTP server has a STEPLIB. If a user exit is not found, processing proceeds as though a return code of 0 was received from the user exit call.

A user exit is passed the address of a parameter list in register 1. The parameter list is a series of pointers to values. The first word of the parameter list always points to the return code. If the user exit sets the return code to 0, processing continues as normal. If the return code is not 0, authorization is denied and the user receives a negative reply indicating that the command has failed. Upon entry, the return code is 0, so a correct return can be indicated by leaving the return code alone.

The second word of the parameter list always points to a word containing the number of parameters that follow. This helps handle future releases that might increase the number of parameters in these parameter lists.

The remainder of the parameter list points to values that the FTP user exit uses in its processing.

Because the FTPCHKIP user exit is loaded at FTP daemon initialization time, if you want the server to use a new version of your exit routine, you need to recycle the FTP server (stop and start it). If you are debugging a user exit routine, you should have a test version of a server to work with so that you can stop and start without

affecting other users. You can do that by putting a PORT parameter in the EXEC statement of the FTP JCL; for example, `PARMS='PORT 1073'`. To connect to this server:

```
FTP nodename 1073
```

You can use any non-well-known number as a port number for your test FTP server.

Note: You cannot use the System Programming C Facilities for the user exits.

The following describes the 4 user exits:

The FTCHKIP User Exit

FTCHKIP is called at the initial stage of logon, or whenever the user issues an OPEN command to open a new connection. The IP and PORT addresses of the local host and remote hosts are passed to the user exit. The user exit can use them to determine if the remote host's control connection should be canceled. The message `421 User Exit rejects open for connection` is sent to the user if the connection is denied. The following parameter list is passed to FTCHKIP.

Offset Value

+0 Pointer to the word with the return code

+4

Pointer to a word containing the number 4 (number of following parameters)

+8 Pointer to the fullword remote IP address

+12 Pointer to the halfword remote port number

+16 Pointer to the fullword local IP address

+20 Pointer to the halfword local port number

The FTCHKPWD User Exit

FTCHKPWD is called just after the user enters the password. The exit is passed the following information: the user ID, password of the user that has just logged on, and a userdata string. The exit has the option of rejecting the logon. The message `530 User Exit rejects logon by 'xxxxx'` is sent to the user if the logon is denied. The following parameter list is passed to FTCHKPWD.

Offset Value

+0 Pointer to the word with the return code

+4

Pointer to a word containing the number (3) (number of following parameters)

+8 Pointer to the 8-byte user ID that is logging on

+12 Pointer to the 8-byte password of user that is logging on

+16 Pointer to the string containing the 2 byte length field followed by the user data.

The FTCHKCMD User Exit

FTCHKCMD is called whenever the user enters a command to execute (such as GET, PUT, or any other FTP command). The user exit is passed the user ID, the command, and the command parameters. The exit has the option of not permitting the execution of the command. The message `500 User Exit denies Userid xxxxx from using Command yyy` is sent to the user if the command is denied. The following parameter list is passed to FTCHKCMD.

Offset Value

+0 Pointer to the word with the return code

+4

Pointer to a word containing the number 3
(number of following parameters)

+8 Pointer to the 8-byte user ID that is logged on

+12 Pointer to the 8-byte command being entered

+16 Pointer to a string containing arguments after the command. The first halfword of the string contains the number of characters that follow.

The FTCHKJES User Exit

FTCHKJES is called if the server is in FILETYPE=JES mode and the client tries to submit a job. The user ID and the job being submitted are passed to the exit. The exit can allow or refuse the job to be submitted to the JES internal reader. For example, the exit can look for a USER= parameter on the JOB statement and check it against the client's user ID. The message `550 User Exit refuses this job to be submitted by userid` is sent to the user if the remote job submission is denied. The following parameter list is passed to FTCHKJES.

Offset Value

+0 Pointer to the word with the return code

+4

Pointer to a word containing the number 8
(number of following parameters)

+8 Pointer to the 8 character user ID that is logged on

+12 Pointer to the buffer containing the current logical record being submitted

+16 Pointer to a word with the number of bytes in the buffer

+20 Pointer to a word containing the JES LRECL being used

+24 Pointer to a word containing the logical record number

+28 Pointer to a word containing the unique client ID

+32

Pointer to a word containing the JES RECFM
(0 for fixed, 1 for variable)

+36 Pointer to a word containing the JES user exit anchor (One possible use of this anchor is to provide the exit routine with a location to store the address of a persistent storage area for handling multiple calls.)

The return code word is initialized to 0, so the user exit can return without changing it if there is a correct return code. Any other return code denies access to the resource in question.

Notes:

1. FTCHKIP has been placed before the user logs on, and if access is denied, the user receives a message and then the control connection is severed. This message comes at a point when most clients expect to continue with the logon process by sending the user ID and password. Even though it is possible that some FTP clients might not expect a 421 message at this point, it is the most appropriate place for this exit.
2. MVS follows the MVS search order to load the FTP exit routines. If you are not using the user exit facility, put a dummy user exit load module in the first library in the MVS search order. This prevents someone from putting in their own module in a library later in the concatenation sequence. This also increases the need to have that library protected using SAF.

Step 7: Specify Statements in TCPIP.DATA

The FTP server gets certain operating parameters from the statements in the TCPIP.DATA data set. This data set has statements that set the TCP/IP client system parameters. Table 10 shows the statements which specifically affect the FTP server. For the search order used for the TCPIP.DATA data set, see "Configuration Data Sets" on page 26.

Summary of FTP Server TCPIP.DATA Statements

Table 10. Summary of FTP Server TCPIP.DATA Statements

Statement	Description	Page
DATASETPREFIX	Set the default high-level qualifier for configuration data sets	104
DOMAINORIGIN	Specify the domain origin that is appended to the host name to form the fully qualified domain name for a host	105
HOSTNAME	Specify the TCP host name of the OS/390 server	106
LOADDBCSTABLES	Tell FTP which DBCS translation tables can be loaded	106
MESSAGECASE	Specify case translation for the FTP server, PING client, oping, onetstat, orouted, and osnmpd	108
TCPIPJOBNAME	Specify the member name of the cataloged procedure used to start the TCPIP address space	112

The following is an example of the statements that affect FTP in the TCPIP.DATA data set:

```
HostName MVS1
DOMAINORIGIN IDD.RALEIGH.IBM.COM
LOADDBCSTABLES TCHINESE
MESSAGECASE LOWER
TCPIPJOBNAME TCPIP
DATASETPREFIX TCP.PROD
```

See Chapter 4, "Defining the TCP/IP Client System Parameters" on page 99 for detailed information about the TCPIP.DATA data set.

Step 8: Install the SQL Query Function and Access the DB2 Modules

To use the FTP server to do SQL queries, bind the DBRM called EZAFTPMQ to the plan to be used by the FTP server, and grant execution privilege for that plan to PUBLIC. (The name of the plan can be specified by the DB2PLAN keyword in FTP.DATA or defaulted to EZAFTPMQ.) This FTP facility only performs SELECT operations on the DB2 tables. It does not perform UPDATE, INSERT, or DELETE.

Note: To use the MVS FTP client do SQL queries, bind the DBRM called MVPSQL. Refer to “Using the FTP Client to do SQL Queries” on page 152.

The following sample job is provided in the FTOEBIND member of the SEZAINST data set.

```
//FTPSETUP JOB FTPSETUP,
//           CLASS=A,
//           NOTIFY=&SYSUID
//*************************************************************
//*
//*   TCP/IP for MVS
//*   File name:              tcpip.SEZAINST(FTOEBIND)
//*   SMP/E distribution name:   EZAFTPAB
//*
//*       5645-001 5655-HAL (C) Copyright IBM Corp. 1997.
//*       Licensed Materials - Property of IBM
//*       This product contains "Restricted Materials of IBM"
//*       All rights reserved.
//*       US Government Users Restricted Rights -
//*       Use, duplication or disclosure restricted by
//*       GSA ADP Schedule Contract with IBM Corp.
//*       See IBM Copyright Instructions.
//*
//*  This JCL binds the EZAFTPMQ DBRM to the specified
//*  DB2 subsystem and allows execution of the
//*  EZAFTPMQ plan by PUBLIC.
//*
//*  This MVS OE FTP server uses this plan.
//*
//*  NOTE:  To access DB2 through the FTP Client you must also
//*  bind the MVPSQL DBRM (see FTCBIND sample).
//*
//*  Usage notes:
//*
//*    1.  You must execute this job from a user ID that has
//*        the authority to bind the EZAFTPMQ plan.
//*
//*    2.  Change the STEPLIB DD statement in the FTPBIND and
//*        FTPGRANT steps to reflect the DB2 DSNLOAD data set.
//*
//*    3.  Change the DB2 sybsystem name in the FTPBIND and
//*        FTPGRANT steps from SYSTEM(xxx) to the
//*        installation defined DB2 subsystem name.
//*
//*    4.  Change the library parameter in the FTPBIND step from
//*        TCPIP.SEZADBRM to the installation defined TCPIP
```

```
//*        SEZADBRM library.
//*
//*    5.  Change the plan name in the FTPGRANT step from
//*        DSNTIAYY to reflect the plan associated with the
//*        program DSNTIAD.
//*
//*    6.  Change the library parameter in the FTPGRANT step
//*        from xxxxxx.RUNLIB.LOAD to reflect the library
//*        where the DSNTIAD program resides.
//*
//*    7.  If you changed the default plan name used by the
//*        OE FTP server (by using DB2PLAN keyword in FTP.DATA),
//*        change the plan name in the FTPBIND and FTPGRANT
//*        steps from EZAFTPMQ to the name specified by the
//*        DB2PLAN keyword in your FTP.DATA data set.
//*
//*************************************************************
//FTPBIND  EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DSN=xxxxxx.DSNLOAD,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSTSIN  DD   *
 DSN  SYSTEM(xxx)
 BIND ACQUIRE(USE) -
      ACTION(REPLACE) -
      CACHESIZE(1024) -
      CURRENTDATA(NO) -
      EXPLAIN(NO) -
      ISOLATION(CS) -
      LIBRARY('TCPIP.SEZADBRM') -
      MEMBER(EZAFTPMQ) -
      NODEFER(PREPARE) -
      PLAN(EZAFTPMQ) -
      RELEASE(COMMIT) -
      VALIDATE(RUN) -
      RETAIN
 END
//*
//FTPGRANT EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DSN=xxxxxx.DSNLOAD,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSTSIN  DD   *
 DSN  SYSTEM(xxx)
 RUN  PROGRAM(DSNTIAD) -
      PLAN(DSNTIAYY) -
      LIBRARY('xxxxxx.RUNLIB.LOAD')
 END
//SYSIN    DD   *
 GRANT EXECUTE ON PLAN EZAFTPMQ TO PUBLIC;
//*
```

Using the FTP Client to do SQL Queries

To use the MVS FTP client do SQL queries, bind the DBRM called MVPSQL.

The following sample job is provided in the FTCBIND member of the SEZAINST data set.

```
//FTCSETUP JOB FTCSETUP,
//            CLASS=A,
//            NOTIFY=&SYSUID
//*****************************************************************
//**                                                             **
//**  FILE NAME:                tcpip.SEZAINST(FTCBIND)          **
//**                                                             **
//**  SMP/E DISTRIBUTION NAME:  EZAFTCBI                         **
//**                                                             **
//**  THIS JCL CAUSES THE BIND OF THE MVPSQL DBRM TO THE         **
//**  SPECIFIED DB2 SUBSYSTEM AND ALLOWS EXECUTION OF THE MVPSQL **
//**  PLAN BY PUBLIC.                                            **
//**                                                             **
//**  THIS PLAN IS USED BY THE FTP CLIENT.                       **
//**                                                             **
//**  NOTE - TO ACCESS DB2 THROUGH THE FTP SERVER YOU MUST       **
//**         BIND THE EZAFTSMQ DBRM (SEE FTSBIND EXAMPLE)        **
//**                                                             **
//**  USAGE NOTES:                                               **
//**                                                             **
//**    1.  THIS JOB MUST BE EXECUTED FROM A USER ID THAT HAS    **
//**        THE AUTHORITY TO BIND THE MVPSQL PLAN.               **
//**                                                             **
//**    2.  CHANGE THE STEPLIB DD STATEMENT IN THE FTPBIND AND   **
//**        FTPGRANT STEPS TO REFLECT THE DB2 DSNLOAD DATASET.   **
//**                                                             **
//**    3.  CHANGE THE DB2 SUBSYSTEM NAME IN THE FTPBIND AND     **
//**        FTPGRANT  STEPS FROM "SYSTEM(XXX)" TO THE            **
//**        INSTALLATION DEFINED DB2 SUBSYSTEM NAME.             **
//**                                                             **
//**    4.  CHANGE THE LIBRARY PARAMETER IN THE FTPBIND STEP     **
//**        FROM 'TCPIP.SEZADBRM' TO THE INSTALLATION DEFINED    **
//**        TCPIP SEZADBRM LIBRARY.                              **
//**                                                             **
//**    5.  CHANGE THE PLAN PARAMETER DEFINED IN THE RUN         **
//**        STATEMENT IN THE FTPGRANT STEP TO REFLECT THE        **
//**        DB2 RELEASE LEVEL INSTALLED ON YOUR SYSTEM           **
//**        (E.G. DSNTIA31)                                      **
//**                                                             **
//**    6.  CHANGE THE LIBRARY PARAMETER IN THE FTPGRANT STEP    **
//**        FROM "XXXXXX.RUNLIB.LOAD" TO REFLECT THE LIBRARY     **
//**        WHERE THE DSNTIAD PROGRAM RESIDES.                   **
//**                                                             **
//*****************************************************************
//FTPBIND  EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DSN=XXXXXX.DSNLOAD,DISP=SHR
//SYSTSPRT DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SYSTSIN  DD   *
 DSN  SYSTEM(XXX)
 BIND ACQUIRE(USE) -
```

```
      ACTION(REPLACE) -
      CACHESIZE(1024) -
      CURRENTDATA(NO) -
      EXPLAIN(NO) -
      ISOLATION(CS) -
      LIBRARY('TCPIP.SEZADBRM') -
      MEMBER(MVPSQL) -
      NODEFER(PREPARE) -
      PLAN(MVPSQL) -
      QUALIFIER(SYSADM) -
      RELEASE(COMMIT) -
      VALIDATE(RUN) -
      RETAIN
 END
//*
//FTPGRANT EXEC PGM=IKJEFT01,DYNAMNBR=20
//STEPLIB  DD   DSN=XXXXXX.DSNLOAD,DISP=SHR
//SYSTSPRT  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSTSIN   DD  *
 DSN SYSTEM(XXX)
 RUN PROGRAM(DSNTIAD) -
     PLAN(DSNTIAXX) -
     LIBRARY('XXXXXX.RUNLIB.LOAD')
 END
//SYSIN  DD  *
 GRANT EXECUTE ON PLAN MVPSQL TO PUBLIC;
//*
```

Accessing DB2 Modules

The FTP server or client loads 3 DB2 modules into storage to perform an SQL query. These modules are:

- DSNALI
- DSNHLI2
- DSNTIAR

The modules are usually found in the DB2 load library with the suffix DSNLOAD. The DB2 administrator or system programmer should add the DSNLOAD library to the LINKLIST to ensure that FTP has access to this library.

Another way to ensure access is to add the DSNLOAD library to the FTP STEPLIB. For the FTP server this means that the JCL that is used to start the FTP server has a STEPLIB DD statement referring to the DSNLOAD library or, if the FTP daemon is started from the OE shell, that the STEPLIB environment variable is set. For the FTP client, this means that a TSO CLIST must allocate the DSNLOAD library as the STEPLIB.

If the FTP client is to be run from a batch job to perform SQL queries, the DSNLOAD library must be added to the STEPLIB DD statement for the batch job.

Usage Notes:

To allow FTP access to multiple levels of DB2, link to the libraries that contain the lowest level of DB2 to be accessed.

Step 9: Update /etc/syslog.conf for the FTP Server

The daemon.*priority* entries in /etc/syslog.conf determine where FTP messages and trace entries are written. The FTP server issues info, warning, and error messages. All trace entries are written with debug priority. To direct trace entries (and all messages) to /tmp/daemon.trace, include the following in /etc/syslog.conf:

```
daemon.debug    /tmp/daemon.trace
```

For information on syslogd, see Appendix C, “Description of Syslog Daemon (syslogd)” on page 259.

Security Considerations for the FTP Server

Consider the following for security:

- Userids

  To log into the OE FTP server, a userid must have an OMVS uid.

- The FTPD cataloged procedure must be:
  - Defined to the security program
  - Added to the RACF started class facility or the started procedures table.

- Terminal Access

  The terminal ID passed from FTP to RACF is an 8-byte hexadecimal character string containing an IP address. RACF interprets this as a terminal logon address and rejects it if it is not previously defined. For example, the IP address 163.97.227.17 is translated to X'A361E311'.

  Therefore, if the SETROPTS TERMINAL(NONE) setting is used in RACF, you must define profiles for the IP addresses in class TERMINAL to avoid problems when trying to FTP to MVS. You must translate all the IP addresses of any clients connecting to FTP servers to hexadecimal character strings and add them to the class TERMINAL.

  To allow access by all addresses starting with “163,” define a profile for all addresses in the 163.97.227 subnet:

  ```
  RDEFINE TERMINAL A361E3* UACC(READ)
  ```

  If your RACF SETROPTS options are TERMINAL(READ), all terminals are allowed access to your system, and you do not have to add extra resource definitions to your RACF data base.

For more information, see *OS/390 OpenEdition Planning* and the *OS/390 Security Server (RACF) Security Administrator's Guide*.

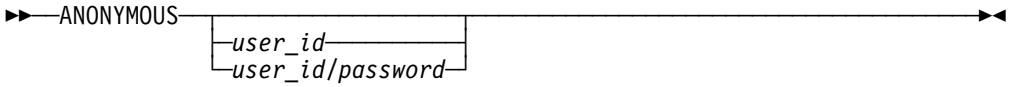
FTP.DATA Data Set Statements

This section covers in detail the statements you can use in the FTP.DATA data set.

ANONYMOUS Statement

Use the ANONYMOUS statement to allow a remote user to issue USER ANONYMOUS without supplying a logon password. Specifying ANONYMOUS makes your universally permitted data sets accessible to all users on the TCP/IP network.

Syntax



Parameters

user_id
The name of the user ID to be used when ANONYMOUS is entered as the user ID. When a remote user enters ANONYMOUS as a user ID, the FTP server treats the login request as though the specified *user_id* was entered instead of ANONYMOUS. The user will be prompted for the password to *user_id* and, if the user enters the correct password, the user will be logged in as the specified *user_id*.

If you are using RACF, the system will build a user Accessor Environment Element (ACEE) and the ANONYMOUS user will have access to any resources available to the specified user ID.

user_id/password
The name of the user ID and password to be used when ANONYMOUS is entered as a user ID. When a remote user enters ANONYMOUS as the user ID, the FTP server treats the login request as though the specified *user_id* was entered instead of ANONYMOUS. The FTP server automatically provides the *password* for the specified *user_id* and the user will be logged in as the specified *user_id*. If you are using RACF, the system will build the user ACEE for the specified *user_id* and the ANONYMOUS user will have authorized access to the same resources as the specified *user_id*.

Examples

Allow a remote user to enter ANONYMOUS as a user ID and be connected to the server system with the user ID of TERMABC:

```
ANONYMOUS TERMABC/ILLBBACK
```

Usage Notes

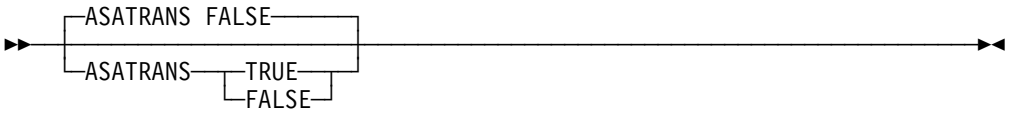
- If you enter the ANONYMOUS statement without a user ID and you are using RACF, the system builds a generic user ACEE and the user will have access to all universally permitted data sets.
- If no userid is specified on the ANONYMOUS statement, the userid ANONYMOU must be defined to RACF and OMVS. If a userid is specified on the ANONYMOUS statement, that userid must be defined.
- There is no default for ANONYMOUS. If you do not include the ANONYMOUS statement in FTP.DATA, then anonymous user login will not be allowed.

ASATRANS Statement

Use the ASATRANS statement to control how the server handles ASA file transfers. Choose either to have the control characters converted by the C runtime during a file transfer or to transfer them without conversion.

The conversion process is described in the *IBM C/370 Programming Guide* in the chapter on ASA Text Files.

Syntax



Parameters

TRUE
Characters in column 1 of the file being transferred are converted to C control character sequences.

FALSE
Characters in column 1 of the file being transferred are not converted. This is the default.

Examples

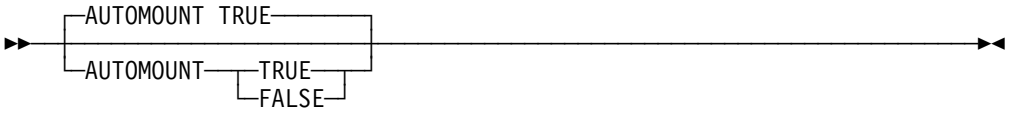
Convert characters in column 1 of the file being transferred:

```
ASATRANS TRUE
```

AUTOMOUNT Statement

Use the AUTOMOUNT statement to permit DASD volumes that are not mounted to be automatically mounted.

Syntax



Parameters

TRUE
Permits DASD volumes that are not mounted to be automatically mounted. This is the default.

FALSE
Prevents DASD volumes that are not mounted from being automatically mounted.

Examples

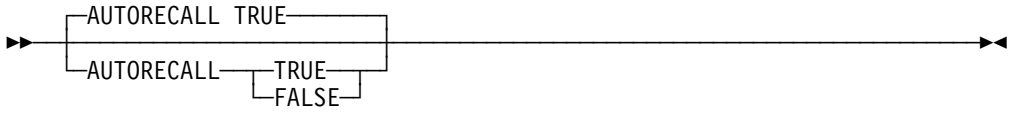
Mount DASD volumes that are not already mounted automatically:

```
AUTOMOUNT TRUE
```

AUTORECALL Statement

Use the AUTORECALL statement to specify whether data sets which have been migrated by a storage manager, such as HSM, will be recalled automatically.

Syntax



Parameters

TRUE
Permits data sets migrated by the storage manager, such as HSM, to be recalled automatically. This is the default.

FALSE
Prevents migrated data sets from being recalled automatically.

Examples

Recall migrated HSM files automatically:

```
AUTORECALL TRUE
```

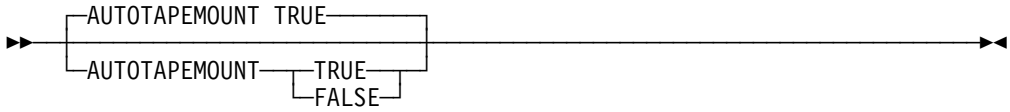
Usage Notes

- Migrated data sets can still be deleted even though you specify `FALSE`.
- Partitioned data set members require the whole data set to be recalled.

AUTOTAPEMOUNT Statement

Use the AUTOTAPEMOUNT statement to specify whether tapes that are not already mounted are to be automatically allocated and mounted.

Syntax



Parameters

TRUE
Permits tapes that are not mounted to be automatically allocated and mounted. This is the default.

FALSE
Prevents tapes that are not mounted from being automatically allocated and mounted.

Examples

Automatically mount tape volumes that are not already mounted:

```
AUTOTAPEMOUNT TRUE
```

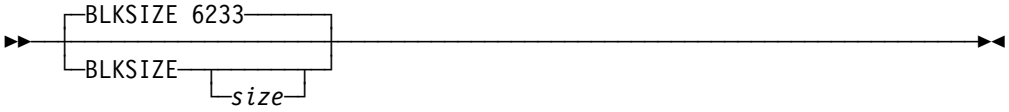
Do not automatically mount tape volumes that are not already mounted.

```
AUTOTAPEMOUNT FALSE
```

BLKSIZE Statement

Use the BLKSIZE statement to specify the block size of newly allocated data sets.

Syntax



Parameters

size
Specifies the block size of newly allocated data sets. The valid range is 0 through 32760. A value of 0 (or no value) for block size allows the block size from a model DCB data set or SMS dataclass to be used. The default block size is 6233.

Examples

Set block size to 6144 bytes:

```
BLKSIZE 6144
```

Specify 0 (or specify no value) for blocksize to allow the blocksize from a model DCB data set or SMS dataclass to be used:

```
BLKSIZE 0
```

or

```
BLKSIZE
```

Usage Notes

- If you specify the BLKSIZE statement with a size of 0 or without a *size*, FTP will not specify the block size when allocating new data sets.
- You should use the BLKSIZE statement without a *size* if you have specified the DATACLASS statement and the block size from the SMS data class is to be used.
- You can also specify this statement as BLOCKSIZE.

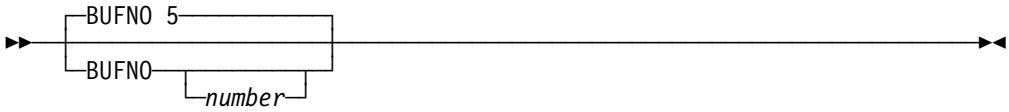
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161
- “DCBDSN Statement” on page 163

BUFNO Statement

Use the BUFNO statement to specify the number of access method buffers that are used when data is read from or written to a data set.

Syntax



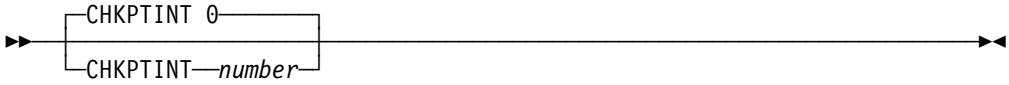
Parameters

number
Specifies the number of buffers allocated. The valid range is 1 through 255. The default is 5.

CHKPTINT Statement

Use the CHKPTINT statement to specify the checkpoint interval when the FTP server is the sending site in a file transfer request.

Syntax



Parameters

number
Used to determine when a restart marker is transmitted. The marker is transmitted after the specified number of records are sent.

If *number* is set to zero, then no checkpointing occurs and no marker blocks are transmitted. The default is zero.

Examples

To send a restart marker of every 100 000 records:

```
CHKPTINT 100000
```

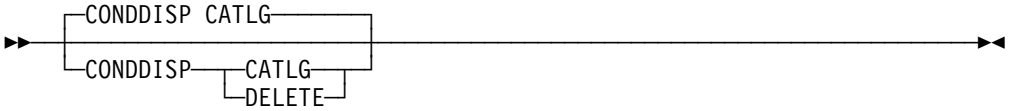
Usage Notes

If a non-zero value is coded for CHKPTINT, checkpoint markers are sent to each client who uses EBCDIC blockmode or EBCDIC compress mode during data set retrieval. If some of those clients do not support the restart marker, you can accept the default value of 0 on this statement and, instead, set the checkpoint interval for an individual client with the FTP SITE command. See *TCP/IP for MVS: User's Guide* for more information on the SITE command.

CONDDISP Statement

Use the CONDDISP statement to keep and catalog or delete a data set when an FTP file transfer ends prematurely.

Syntax



Parameters

CATLG
Specifies that a data set is kept and cataloged when an FTP file transfer ends prematurely. This is the default.

DELETE
Specifies that a data sets is deleted when a file transfer ends prematurely.

Examples

Specify that a data set is deleted when a file transfer ends prematurely:

```
CONDDISP DELETE
```

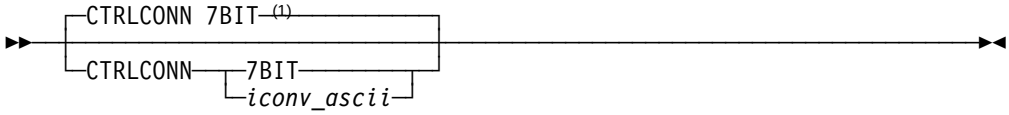
Usage Notes

- DELETE is ignored if the file transfer ended prematurely because the FTP server was stopped.
- DELETE is ignored if the server receives a checkpoint marker.

CTRLCONN Statement

This statement defines the ASCII code page to be used for the control connection.

Syntax



Note:
[1] 7BIT is the default if CTRLCONN is not used and no TCPXLBIN data set is found.

Parameters

7BIT
Indicates 7-bit ASCII is to be used

iconv_ascii
A name recognized by iconv to indicate an ASCII code page.

Examples

```
CTRLCONN IBM-850
```

Usage Notes

7BIT or an *iconv_ascii* name can be entered in lowercase or uppercase.

To see the search order, see “OE FTP Code Page Conversion” on page 192.

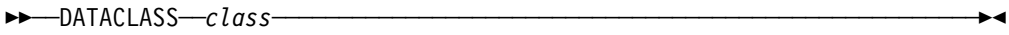
Related Topics

For the code pages supported, see code set converters in the *OS/390 C/C++ Programming Guide*.

DATACLASS Statement

Use the DATACLASS statement to specify the SMS-managed data class as defined by your organization for the FTP server.

Syntax



Parameters

class
The SMS-managed data class as defined by your organization. There is no default.

Examples

Use the SMS data class SMSDATA when allocating new data sets:

```
DATACLASS SMSDATA
```

Usage Notes

If you specify any of the following FTP.DATA statements, or let them default, the values specified or defaulted will override the value specified in the SMS DATACLASS:

- BLKSIZE
- DIRECTORY
- LRECL
- PRIMARY
- RECFM
- RETPD
- SECONDARY

If you specify the DCBDSN statement, the LRECL, RECFM, BLKSIZE, and RETPD (if specified) of the DCBDSN data set will override the values specified in the SMS DATACLASS. To prevent these keywords from overriding the values specified in the SMS DATACLASS, specify them with no keyword values.

If you specify the MGMTCLASS statement, and the requested management class specifies a retention period, the RETPD value of the management class might override the RETPD value of DATACLASS.

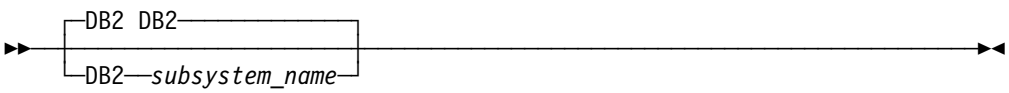
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “BLKSIZE Statement” on page 158
- “DCBDSN Statement” on page 163
- “DIRECTORY Statement” on page 165
- “LRECL Statement” on page 169
- “MGMTCLASS Statement” on page 170
- “PRIMARY Statement” on page 171
- “RECFM Statement” on page 172
- “RETPD Statement” on page 174
- “SECONDARY Statement” on page 176

DB2 Statement

Use the DB2 statement to specify the name of the DB2 subsystem.

Syntax



Parameters

subsystem_name
The name of the DB2 subsystem. The default name is DB2.

Examples

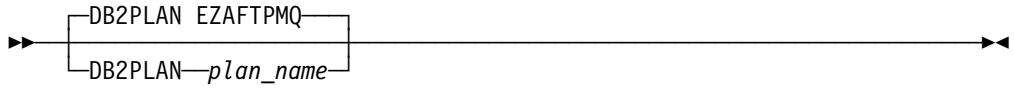
Set the DB2 subsystem name to DB2X:

```
DB2 DB2X
```

DB2PLAN Statement

Use the DB2PLAN statement to specify the DB2 plan to be used by the FTP server.

Syntax



Parameters

plan_name
The name of the DB2 plan bound in the DB2 subsystem.

Examples

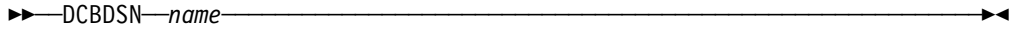
Set the plan name to FTPPLAN:

```
DB2PLAN   FTPPLAN
```

DCBDSN Statement

Use the DCBDSN statement to specify an MVS data set to be used as a model for allocation of new data sets.

Syntax



Parameters

name
The name of the data set that will be used as a model for allocation of new data sets created with a STOR or MKDIR command. This data set name must be a fully-qualified MVS data set name; HFS file names are not allowed. There is no default.

Examples

Use model.dcb as the model data set for allocation and specify RECFM, LRECL, and BLKSIZE with no parameters to allow the attributes from the model DCB to be used:

```
DCBDSN model.dcb
BLKSIZE
LRECL
RECFM
```

BLKSIZE and LRECL can also be specified with a value of 0 to allow the attributes from the model DCB to be used:

```
DCBDSN model.dcb
BLKSIZE 0
LRECL 0
RECFM
```

Usage Notes

If specified or defaulted, the following FTP.DATA statements or SITE command parameters will override the DCB values from the model data set:

BLKSIZE
LRECL
RECFM
RETPD

If you specify the MGMTCLASS statement, the retention period from the model data set can be overridden by the retention period specified by the SMS management class.

When using a model DCB at the server, SENDSITE must be toggled off at the client. Otherwise, the SITE information that is sent automatically by the client will override the value provided by the model DCB.

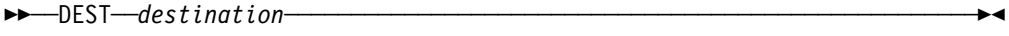
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “BLKSIZE Statement” on page 158
- “LRECL Statement” on page 169
- “MGMTCLASS Statement” on page 170
- “RECFM Statement” on page 172
- “RETPD Statement” on page 174

DEST Statement

Use the DEST statement to specify the NJE destination to which the files are routed when you enter a STOR command. Using the DEST statement allows you to send data sets to other users on machines that are connected on a Network Job Entry (NJE) network rather than storing them at the server.

Syntax



Parameters

destination
The NJE destination to which the files are routed when you enter a PUT command. The format for *destination* should be one of the following:

- userID@nodeID
- nodeID.userID
- nodeID
- DestID

There is no default.

Examples

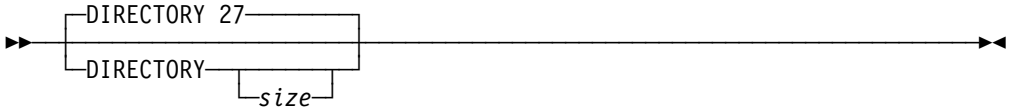
Send files to user USER14 at system MVS1 on a STOR command:

```
DEST USER14@MVS1
```

DIRECTORY Statement

Use the DIRECTORY statement to specify the number of directory blocks to be allocated for the directory of a PDS.

Syntax



Parameters

size
The number of directory blocks to be allocated for the directory of a PDS. The valid range is 1 to 16777215 blocks (the operating system maximum). The default is 27.

Examples

Allocate a PDS with 15 directory blocks:

```
Directory 15
```

Specify DIRECTORY with no value to allow the directory information from an SMS dataclass to be used:

```
DIRECTORY
```

Usage Notes

- If you specify no value for the *size*, FTP will not specify the number of directory blocks to be allocated for the directory of a PDS.
- You should specify no value for the *size* if the DATACLASS statement is specified and the directory from the SMS data class is to be used.

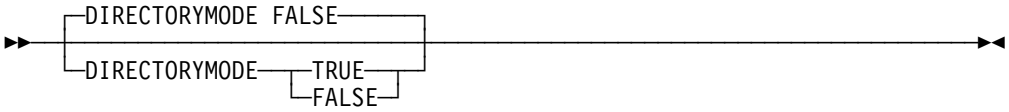
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161

DIRECTORYMODE Statement

Use the DIRECTORYMODE statement to specify whether only the data set qualifier immediately below the current directory is treated as an entry in the directory or if all the data set qualifiers below the current directory are treated as entries in the directory.

Syntax



Parameters

TRUE
Specifies that only the data set qualifier immediately below the current directory is treated as an entry in the directory.

FALSE
Specifies that all the data set qualifiers below the current directory are treated as entries in the directory. This is the default.

Examples

Use all qualifiers (Datasetmode):

```
DirectoryMode FALSE
```

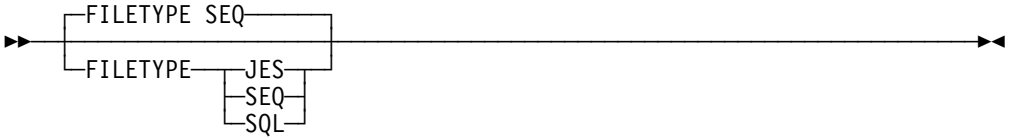
Usage Notes

In directory mode, only the data set qualifier immediately below the current directory is the only one used by the MPUT, MGET, LS, and DIR subcommands.

FILETYPE Statement

Use the FILETYPE statement to specify the mode of operation of the server.

Syntax



Parameters

JES
Remote job submission

SEQ
Sequential or partitioned data sets. This is the default.

SQL
SQL query function

Examples

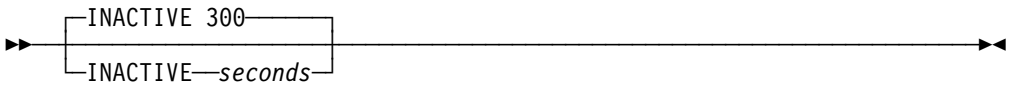
Set the operational mode to SQL:

```
Filetype SQL
```

INACTIVE Statement

Use the INACTIVE statement to set the inactivity timer to a specified number of seconds. Any client control connection which is inactive for the amount of time specified on this statement is closed by the server.

Syntax



Parameters

seconds
The number of seconds to which the inactivity timer will be set. The valid range is 0 through 86 400. The default is 300.

Examples

Set the inactivity timer to 30 seconds:

```
INACTIVE 30
```

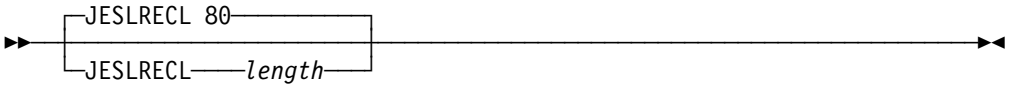
Usage Notes

If you specify 0 seconds, the inactivity timer will be disabled and the control connections will never time out. This value has no effect on the data connections. To specify a timeout value for the data connection, use the KEEPALIVEOPTIONS statement in TCPIP.DATA. Refer to “Step 1: Specify Port and KEEPALIVEOPTIONS Information” on page 133 for details.

JESLRECL Statement

Use the JESLRECL statement to specify the record length of the jobs being submitted.

Syntax



Parameters

length
The record length of the job being submitted. The valid range is 1–254. The default is 80. If you specify *length* as *, FTP uses the length value from the LRECL statement.

Examples

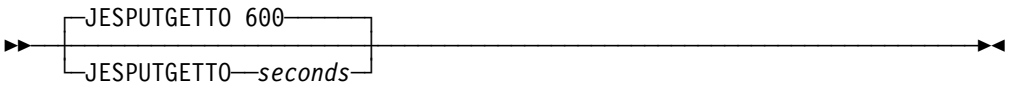
Explicitly set the logical record length for JES jobs to 80:

```
JESLRECL 80
```

JESPUTGETTO Statement

Use the JESPUTGETTO statement to specify the number of seconds of the JES PutGet time-out.

Syntax



Parameters

seconds
The number of seconds of the JES PutGet time-out. The valid range is 0 through 86 400 (24 hours). The default is 600 (10 minutes).

Examples

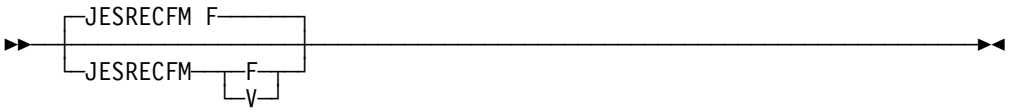
Set the number of seconds of the JES PutGet time-out to 300:

```
JESPUTGETTO 300
```

JESRECFM Statement

Use the JESRECFM statement to specify the record format of the jobs being submitted.

Syntax



Parameters

F
Fixed record length. This is the default.

V
Uses the record format specified on the RECFM statement.

Examples

Use fixed record format:

```
JESRECFM F
```

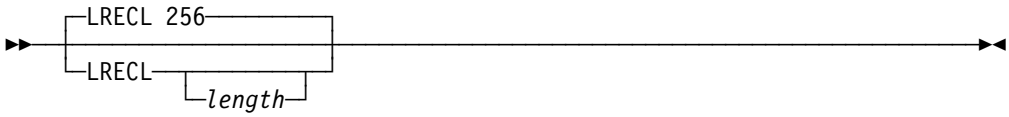
Usage Notes

Only use the value V when running on JES3 systems.

LRECL Statement

Use the LRECL statement to specify the size of the records in a data set.

Syntax



Parameters

length
The size of the records in a data set. The valid range is 0 through 32760. The default is 256.

Examples

Set the logical record length to 128 bytes:

```
LRECL 128
```

Specify no value for LRECL to allow the LRECL of a model DCB data set or SMS dataclass to be used:

```
LRECL
```

Usage Notes

- If you specify no value for *length*, FTP will not specify the size of the records in a data set.
- You should specify no value for *length* if the DATACLASS statement is specified and the LRECL from the SMS data class is to be used, or if the DCBDSN statement is specified and the LRECL from the model data set is to be used.

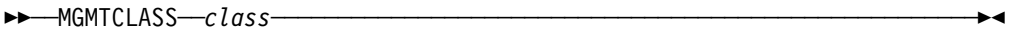
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161
- “DCBDSN Statement” on page 163

MGMTCLASS Statement

Use the MGMTCLASS statement to specify the SMS management class to be assigned to newly allocated data sets.

Syntax



Parameters

class
The SMS management class.

Examples

Set the SMS management class for new data sets to TCPMGMT:

```
MGMTCLASS TCPMGMT
```

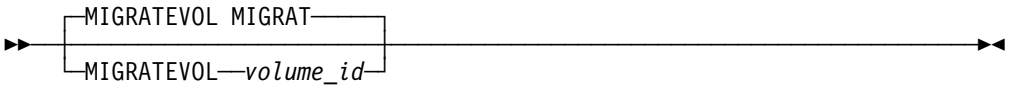
Related Topics

See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.

MIGRATEVOL Statement

Use the MIGRATEVOL statement to specify the volume ID for migrated data sets under the control of a storage management system other than HSM.

Syntax



Parameters

volume_id
The volume ID for migrated data sets. The default volume ID is MIGRAT.

Examples

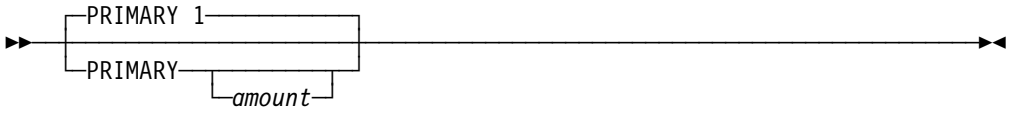
Set the volume ID for migrated data sets to MIGRIX:

```
MIGRATEVOL MIGRIX
```

PRIMARY Statement

Use the PRIMARY statement to specify the amount of tracks, blocks, or cylinders (according to SPACETYPE) for primary allocation.

Syntax



Parameters

amount
The amount of tracks, blocks, or cylinders. The valid range is 1 to 16777215 blocks (the operating system maximum). The default is 1.

Examples

Set the primary allocation to 5 tracks:

```
PRIMARY 5
```

Usage Notes

- If you specify no value for *amount*, FTP will not specify the amount of tracks, blocks, or cylinders for primary allocation.
- You should specify no value for *amount* if the DATACLASS statement is specified and the space allocation from the SMS data class is to be used. If the SMS data class is to be used for space allocation, both the PRIMARY and SECONDARY values must be omitted and the value on the SPACETYPE statement will be ignored.
- For allocating partitioned data sets, *amount* is the amount that will be allocated for the primary extent.
- For allocating sequential data sets, *amount* is the maximum amount that will be allocated for the primary extent. If a lesser amount is needed to hold the data being transferred, only the amount actually needed to hold the data will be allocated.

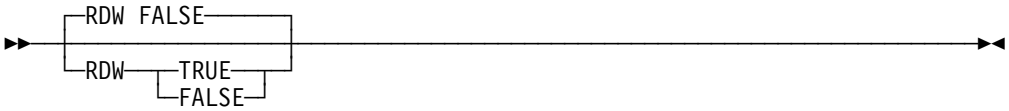
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161
- “SECONDARY Statement” on page 176
- “SPACETYPE Statement” on page 182

RDW Statement

Use the RDW statement to specify if the RDW from variable format data sets should be retained as data.

Syntax



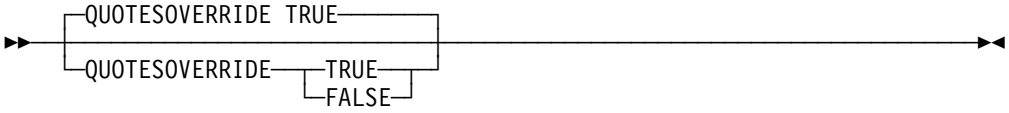
Usage Notes

- If TRUE is specified, RDWs are transferred as part of the data.
- If FALSE is specified, RDWs are discarded when transferring variable format data sets.

QUOTESOVERRIDE Statement

Use the QUOTESOVERRIDE statement to indicate the usage of single quotes appearing at the beginning of, or surrounding, a filename.

Syntax



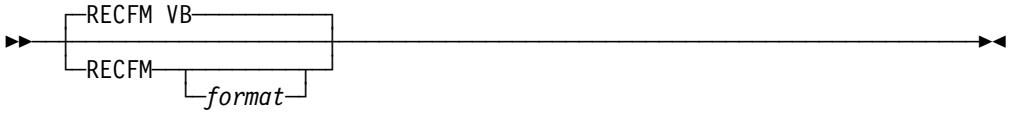
Usage Notes

- If TRUE is specified, then single quotes appearing at the beginning and end of a filename are interpreted to mean that the filename contained inside the single quotes should override the current working directory instead of being appended to the current working directory. This is the way single quotes are currently used in all previous MVS FTP servers, and is the default. Any single quotes inside the beginning and ending quote are treated as part of the filename.
- If FALSE is specified, then a single quote at the beginning of the file name, as well as all other single quotes contained in the filename, will be treated as part of the actual filename. The entire file name, including the leading single quote, will be appended to the current working directory.

RECFM Statement

Use the RECFM statement to specify the record format of a data set.

Syntax



Parameters

format
The record format of a data set. Valid record formats are: F, FM, FA, FS, FSA, FSM, FB, FBM, FBA, FBS, FBSM, FBSA, V, VM, VA, VS, VSM, VSA, VB, VBM, VBA, VBS, VBSA, VBSM, U, UA, and UM. The default record format is VB. The meanings of the record formats are:

Format	Description
A	Records contain ISO/ANSI control
B	Blocked records
F	Fixed record length
M	Records contain machine code control characters
S	Spanned records (if variable), or Standard (if fixed)
U	Undefined record length
V	Variable record length characters

Examples

Use fixed blocked record format:

```
RECFM FB
```

Specify RECFM with no value to allow the RECFM value of a DCB data set or an SMS dataclass to be used:

```
RECFM
```

Usage Notes

- If you specify no value for *format*, no record format will be specified when allocating new data sets.
- You should specify no value for *format* if you specify the DATACLASS statement and the record format from the SMS data class is to be used.

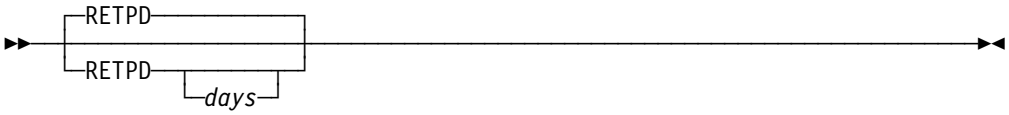
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161
- “DCBDSN Statement” on page 163

RETPD Statement

Use the RETPD statement to specify the number of days that a newly allocated data set should be retained.

Syntax



Parameters

days
The number of days that a newly allocated data set should be retained. The valid range is 0 through 9999. The default is to have no retention period assigned to the data set.

Examples

- Make the new data set expiration date to be 30 days:

  ```
  RETPD 30
  ```

- Use a retention period of 0 days:

  ```
  RETPD 0
  ```

Usage Notes

- If you do not specify the RETPD statement or if you the RETPD statement with no value, no retention period is assigned to newly allocated data sets.
- You should specify no value for *days* if the DATACLASS statement is specified and the retention period from the SMS data class is to be used.
- If you specify 0 for *days*, newly allocated data sets will be assigned a retention period of 0 days. This means that the retention period of the data set will expire on the same day that the data set is created.
- If the SMS data class or DCBDSN model data set have a retention period, this retention period can be overridden to a new retention period. The retention period cannot be overridden to have no assigned retention period.

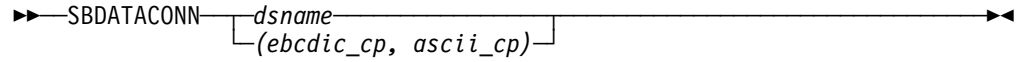
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161
- “DCBDSN Statement” on page 163

SBDATACONN Statement

This statement defines the conversions between EBCDIC and ASCII code pages to be used for data transfer.

Syntax



Parameters

dsname
The fully-qualified name of an MVS data set or HFS file that contains the EBCDIC to ASCII translate tables and the ASCII to EBCDIC translate tables generated by the CONVXLAT utility. For more information on translation tables, see “Using Translation Tables” in the *TCP/IP for MVS: Customization and Administration Guide*.

ebcdic_cp
The name of an EBCDIC code page recognized by iconv

ascii_cp
the name of an ASCII code page recognized by iconv

Examples

```
SBDATACONN (IBM-037, IBM-850)
```

Usage Notes

- The SYSFTSX DD statement, if present, overrides the SBDATACONN statement.
- If both the SYSFTSX DD statement and the SBDATACONN statement are not present, the search order for a TCPXLBIN data set is followed. See “OE FTP Code Page Conversion” on page 192 for this search order. If no TCPXLBIN data set is found, the same conversion established for the control connection is used for single-byte data transfer.
- The dsname must *not* be enclosed in quotes. If quotes appear, they will be treated as part of the name.
- The HFS name is case-sensitive. The MVS name may be entered in any case.
- The length of an HFS name is limited by the record length of the FTP.DATA file because the SBDataconn statement must fit on one line.
- The HFS name cannot start with a left paren (() or contain semicolons (;) or any blanks ( ).
- If you specify SBDATACONN (*ebcdic_cp*, *ascii_cp*), FTP uses the iconv() application programming interface to do translation from EBCDIC to ASCII and ASCII to EBCDIC. The values that you enter on the SBDATACONN statement are used by FTP as parameters to the iconv() interface.

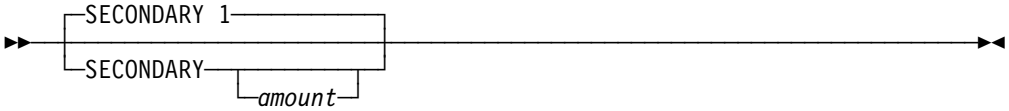
Related Topics

For the code pages supported by iconv, see code set converters in the *OS/390 C/C++ Programming Guide*.

SECONDARY Statement

Use the SECONDARY statement to specify the amount of tracks, blocks, or cylinders (according to SPACETYPE) for secondary allocation.

Syntax



Parameters

amount
The amount of tracks, blocks, or cylinders. The valid range is 0 to 16777215 blocks (the operating system maximum). The default is 1.

Examples

Set the secondary allocation to 2 tracks:

```
SECONDARY 2
```

Usage Notes

- If you specify no value for *amount*, FTP will not specify the amount of tracks, blocks, or cylinders for secondary allocation.
- You should specify no value for *amount* if the DATACLASS statement is specified and the space allocation from the SMS data class is to be used. If the SMS data class is to be used for space allocation, both the PRIMARY and SECONDARY values must be omitted and the value on the SPACETYPE statement will be ignored.

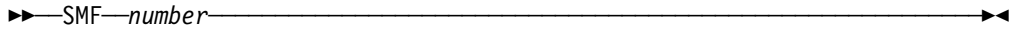
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “DATACLASS Statement” on page 161
- “PRIMARY Statement” on page 171
- “SPACETYPE Statement” on page 182

SMF Statement

Use the SMF statement to specify the default SMF record subtype to be used for all SMF records.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value.

Examples

Set the default SMF record subtype to 70:

```
SMF 70
```

Usage Notes

- This statement defines the default SMF record subtype to be used for any of the following record subtypes that are omitted from the *hlq*.FTP.DATA data set.
  - APPEND
  - DELETE
  - RENAME
  - RETRIEVE
  - STORE
  - STORE UNIQUE
  - Login failure
- If you do not use the SMF statement, then SMF records will not be written for any of the above record subtypes that are omitted.
- If none of the SMF subtype statements are coded in the *hlq*.FTP.DATA data set, then no SMF records are written by the FTP server.

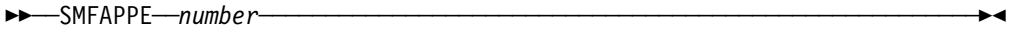
Related Topics

- "SMFAPPE Statement"
- "SMFDEL Statement" on page 178
- "SMFJES Statement" on page 179
- "SMFREN Statement" on page 180
- "SMFRETR Statement" on page 181
- "SMFSQL Statement" on page 181
- "SMFSTOR Statement" on page 182

SMFAPPE Statement

Use the SMFAPPE statement to specify the SMF record subtype to be used for the APPE (APPEND) subcommand.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value, however, if the SMF statement is coded, the value specified for the SMF statement will be used as the default.

Examples

Set the SMF record subtype for APPEND to 71:

```
SMFAPPE 71
```

Usage Notes

If you do not specify the SMFAPPE statement, the *number* on the SMF statement will be used for APPEND records if the SMF statement is specified. If neither the SMF or the SMFAPPE statement is specified, no SMF records will be collected for the APPEND subcommand.

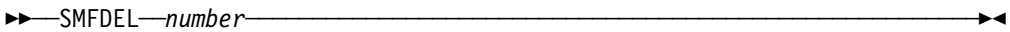
Related Topics

"SMF Statement" on page 176

SMFDEL Statement

Use the SMFDEL statement to specify the SMF record subtype to be used for the DELE (DELETE) subcommand.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value, however, if the SMF statement is coded, the value specified for the SMF statement will be used as the default.

Examples

Set the SMF record subtype for DELETE to 72:

```
SMFDEL 72
```

Usage Notes

If you do not specify the SMFDEL statement, then the *number* on the SMF statement will be used for DELETE records if the SMF statement is specified. If neither the SMF or the SMFDEL statement is specified, no SMF records will be collected for the DELETE subcommand.

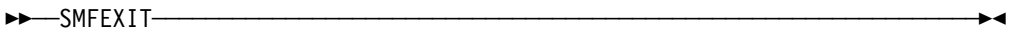
Related Topics

"SMF Statement" on page 176

SMFEXIT Statement

Use the SMFEXIT statement to specify that the user exit routine FTPSMFEX is called before passing the SMF record to SMF.

Syntax



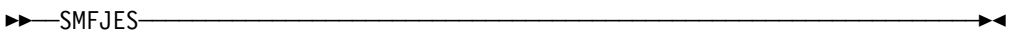
Parameters

None.

SMFJES Statement

Use the SMFJES statement to specify that SMF records are collected when FILETYPE is JES (remote job submission).

Syntax



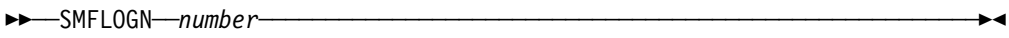
Parameters

None.

SMFLOGN Statement

Use the SMFLOGN statement to specify the SMF record subtype to be used when recording logon failures.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value, however, if the SMF statement is coded, the value specified for the SMF statement will be used as the default.

Examples

Set the SMF record subtype for login failure records to 73:

```
SMFLOGN 73
```

Usage Notes

If you do not specify the SMFLOGN statement, then the *number* on the SMF statement will be used for login failure records if the SMF statement is specified. If neither the SMF or the SMLOGN statement is specified, no SMF records will be collected for login failures.

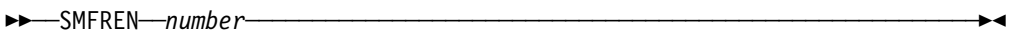
Related Topics

"SMF Statement" on page 176

SMFREN Statement

Use the SMFREN statement to specify the SMF record subtype to be used for the RNFT/RNTO (RENAME) subcommand.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value, however, if the SMF statement is coded, the value specified for the SMF statement will be used as the default.

Examples

Set the SMF record subtype for RENAME records to 74:

```
SMFREN 74
```

Usage Notes

If you do not specify the SMFREN statement, then the *number* on the SMF statement will be used for RENAME records if the SMF statement is specified. If neither the SMF or the SMLOGN statement is specified, no SMF records will be collected for the RENAME subcommand.

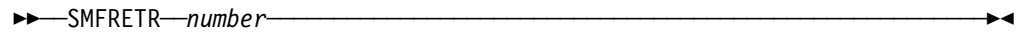
Related Topics

"SMF Statement" on page 176

SMFRETR Statement

Use the SMFRETR statement to specify the SMF record subtype to be used for the RETR (RETRIEVE) subcommand.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value, however, if the SMF statement is coded, the value specified for the SMF statement will be used as the default.

Examples

Set the SMF record subtype for RETRIEVE records to 75:

```
SMFRETR 75
```

Usage Notes

If you do not specify the SMFRETR statement, then the *number* on the SMF statement will be used for RETRIEVE records if the SMF statement is specified. If neither the SMF or the SMLOGN statement is specified, no SMF records will be collected for the RETRIEVE subcommand.

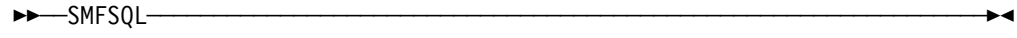
Related Topics

"SMF Statement" on page 176

SMFSQL Statement

Use the SMFSQL statement to specify that SMF records are collected when FILETYPE is SQL (SQL query function).

Syntax



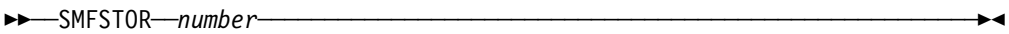
Parameters

None.

SMFSTOR Statement

Use the SMFSTOR statement to specify the SMF record subtype to be used for the STOR (STORE) and STOU (STORE UNIQUE) subcommands.

Syntax



Parameters

number
The SMF record subtype. The valid range is 1 through 255. There is no default value, however, if the SMF statement is coded, the value specified for the SMF statement will be used as the default.

Examples

Set the SMF record subtype for STORE and STORE UNIQUE records to 76:

```
SMFSTOR 76
```

Usage Notes

If you do not specify the SMFSTOR statement, then the *number* on the SMF statement will be used for STORE and STORE UNIQUE records if the SMF statement is specified. If neither the SMF or the SMLOGN statement is specified, no SMF records will be collected for the STORE subcommand.

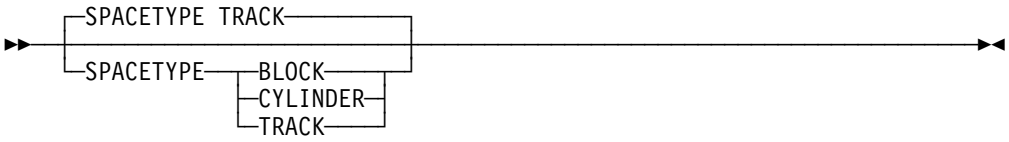
Related Topics

"SMF Statement" on page 176

SPACETYPE Statement

Use the SPACETYPE statement to specify whether newly allocated data sets are allocated in blocks, cylinders, or tracks.

Syntax



Parameters

BLOCK
Use blocks when allocating new data sets.

CYLINDER
Use cylinders when allocating new data sets.

TRACK
Use tracks when allocating new data sets. This is the default.

Examples

Allocate data sets in tracks:

```
SPACETYPE TRACK
```

Usage Notes

If you do not give values on the PRIMARY and SECONDARY statements in order to use the SMS data class, the value on the SPACETYPE statement will be ignored and SMS will determine the spacetype.

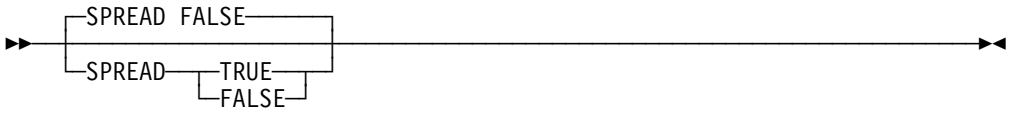
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “PRIMARY Statement” on page 171
- “SECONDARY Statement” on page 176

SPREAD Statement

Use the SPREAD statement to specify whether or not the output is in spreadsheet format when the file type is SQL.

Syntax



Parameters

TRUE
Specifies that the output is in spreadsheet format.

FALSE
Specifies that the output is not in spreadsheet format. This is the default.

Examples

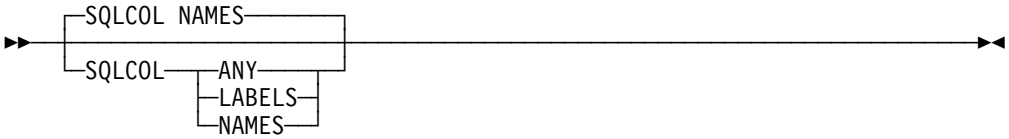
Make the output be in spreadsheet format:

```
SPREAD TRUE
```

SQLCOL Statement

Use the SQLCOL statement to specify the column headings of the output file.

Syntax



Parameters

ANY
Use the label, but if there is no label, the name becomes the column heading.

LABELS
Use the label of the column headings. If any of the columns do not have labels, the server uses COL*number*, where *number* is the column number reading left to right.

NAMES
Use the name of the column headings and ignore the labels. This is the default.

Examples

Use the label of the column headings:

```
SQLCOL LABELS
```

STORCLASS Statement

Use the STORCLASS statement to specify the SMS storage class as defined by your organization for the FTP server.

Syntax

Parameters

class
The SMS class.

Examples

Use the SMS storage class SMSSTOR when allocating new data sets:

```
STORCLASS SMSSTOR
```

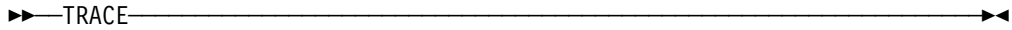
Related Topics

See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.

TRACE Statement

Use the TRACE statement to start tracing for the FTP server. The trace output will be written to syslog.

Syntax



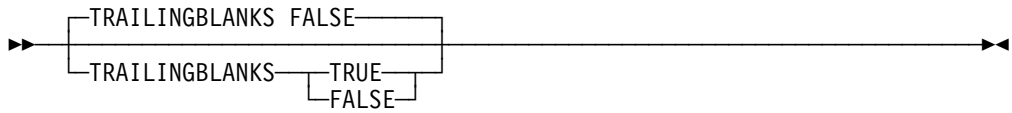
Parameters

None.

TRAILINGBLANKS Statement

Use the TRAILINGBLANKS statement to specify whether trailing blanks in a fixed format data set are transferred when the data set is transferred.

Syntax



Parameters

TRUE
Specifies that the trailing blanks in a fixed format data set are included when the data set is retrieved.

FALSE
Specifies that the trailing blanks in a fixed format data set are not retrieved. This is the default.

Usage Notes

Retrieve the fixed format data set and include trailing blanks:

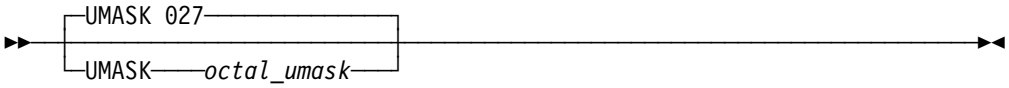
```
TRAILINGBLANKS TRUE
```

UMASK Statement

Use the UMASK statement to define the file mode creation mask.

The file mode creation mask defines which permission bits are NOT to be set on when a file is created. When a file is created, the permission bits requested by the file creation are compared to the file mode creation mask, and any bits requested by the file creation which are disallowed by the file mode creation mask are turned off.

Syntax



Parameters

octal_umask
is the octal umask.

Examples

When a file is created the permission bits for file creation are 666 (-rw-rw-rw-). If the file mode creation mask is 027, the requested permissions and the file mode creation mask are compared:

```
110110110   - 666
000010111   - 027
---------
110100000   - 640
```

When the UMASK is set to 027, the actual permission bits set for a file when it is created will be 640 (-rw-r-----).

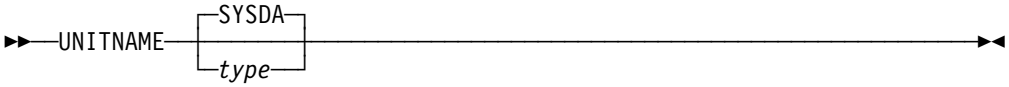
Usage Notes

You cannot use FTP to create HFS files having execute permissions. If you require execute permissions, use the *site chmod* command after the file is created. For more information on *site chmod*, see the *OS/390 TCP/IP OpenEdition User's Guide*.

UNITNAME Statement

Use the UNITNAME statement to specify the unit type for allocation of new data sets.

Syntax



Parameters

type
The type of either direct access or tape devices.

SYSDA
If *type* is not specified, SYSDA is the default.

Examples

- Set the unit type for new data sets to 3380:

  ```
  UNITNAME 3380
  ```

- Set the unit type for new data sets to TAPE:

  ```
  UNITNAME TAPE
  ```

Usage Notes

- If you do not use the UNITNAME statement to specify the *type*, then the unit type used for allocation is the system default unit.
- If the STORCLASS statement is also specified, the SMS storage class might contain settings that will override the UNITNAME *type*.
- It is preferable that you do not use the UNITNAME statement if you are using an SMS storage class.
- The UNITNAME can name a dynamic device.

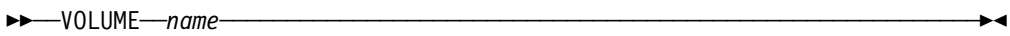
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “STORCLASS Statement” on page 184

VOLUME Statement

Use the VOLUME statement to specify the volume serial number for allocation of new data sets.

Syntax



Parameters

name
The volume serial number. The value specified for *name* is case-sensitive.

Examples

Set the volume name for new data set to WRKLB4:

```
VOLUME WRKLB4
```

Usage Notes

- If you do not use the VOLUME statement to specify the *name*, then the volume serial number used for allocation is the system default volume list.
- If the STORCLASS statement is also specified, the SMS storage class might contain settings that will override the VOLUME *name*.
- It is preferable that you do not use the VOLUME statement if you are using an SMS storage class.

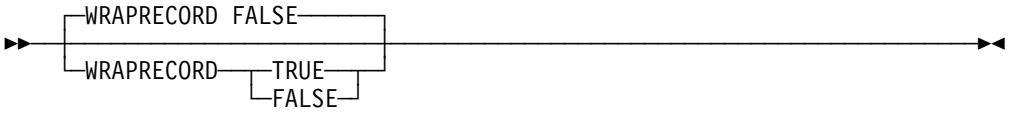
Related Topics

- See “Storage Management Subsystem (SMS)” on page 142 for more information about specifying attributes when allocating new data sets.
- “STORCLASS Statement” on page 184

WRAPRECORD Statement

Use the WRAPRECORD statement to specify whether data will be wrapped to the next record or truncated if no new line character is encountered before the logical record length is reached.

Syntax



Parameters

TRUE
Indicates that data will be wrapped to the next record if no new line character is encountered before the logical record length is reached.

FALSE
Indicates that data will be truncated if no new line character is encountered before the logical record length is reached. This is the default.

Examples

Truncate data if no new line character is encountered before the logical record length is reached:

```
WRAPRECORD FALSE
```

Starting, Stopping, and Tracing the OE FTP Server

The FTP server can be started as a batch job using JCL or from the OE shell. It can also be started automatically when OMVS is started using the /etc/rc file.

The FTP server offers MODIFY command support to start and stop trace dynamically. Use of this support is restricted to the users with MODIFY command privilege.

Starting the OE FTP Server

The FTP server uses the ETC.SERVICES file to determine which port address to use for the FTP control port. If desired, the control port for the FTP server may also be specified using the PORT start option of the FTP server. If the PORT start option is specified when starting the FTP server, this value will override the value specified in the ETC.SERVICES file. If there is no entry for the FTP server in the ETC.SERVICES file, and the PORT start option is not specified, the default ports for the FTP server are port 21 for the control port and port 20 for the data port.

During initialization, the jobname of the server changes from the original jobname to a new jobname:

- The ***original*** jobname is used as the hlq for the initialization and config data sets.
- The ***new*** job name is used for operator STOP and MODIFY commands.

The new job name and process ID are logged in syslogd in message EZYFT41I.

Starting OE FTP from a Batch Job

Use the START command to start the FTP server as a started task:

where:

procname
The name of the FTP cataloged procedure library. See “FTP Server Cataloged Procedure (FTPD)” on page 134 and “Specifying the FTPD Parameters” on page 135.

Starting OE FTP from the OE Shell

From the OE shell, invoke the ftpd module. For example, to start an instance of FTP using port 8097, use the following:

```
/usr/sbin/ftpd   port 8097
```

Note: You must be in superuser to successfully start the FTP daemon from the shell.

Starting OE FTP Automatically

To start the FTP daemon automatically when OMVS is started, add the following to /etc/rc:

```
export _BPX_JOBNAME='FTPD'
/usr/sbin/ftpd <start parameters> &
```

Notes:

- Syslogd should be started before FTP, or all messages and trace entries will appear on the MVS system console.
- If TCP/IP initialization is not complete before FTP is started, the FTP server will be unable to establish a socket and the following message will be sent to syslogd and written to the appropriate HFS file:

  ```
  EZYFT12E socket error: EDC5112I Resource temporarily unavailable
  ```

  The FTP server will continue to try every minute until TCP/IP initialization is complete, at which time FTP initialization can complete. FTP will not recognize a stop command at this stage of its initialization, but you can issue a cancel command.

OE FTP Server Exit Codes

OE FTP uses the following error exit codes:

Exit Code	Explanation
12	Daemon initialization failed, or unable to accept an incoming connection. An EZY message identifying the specific problem is issued to syslog.
24	Client session initialization terminated because the FTP server load module cannot be loaded or executed. Message EZYFT53E is issued to syslog.

Stopping the OE FTP Server

The main OE FTP server process can be stopped either by issuing the MVS STOP command to the new jobname or by issuing an OE shell "kill" command for the process ID. Stopping the main OE FTP Server process will not affect the active session processes. The active session processes will remain active until the client/server session is terminated by an FTP QUIT command or until the client process is stopped.

The individual client session processes cannot be stopped with an MVS STOP command. The client process can only be stopped by issuing an OE shell "kill" command for the client session process. To aid in killing the client session processes, the process ID for the client will be returned in the STAT command output.

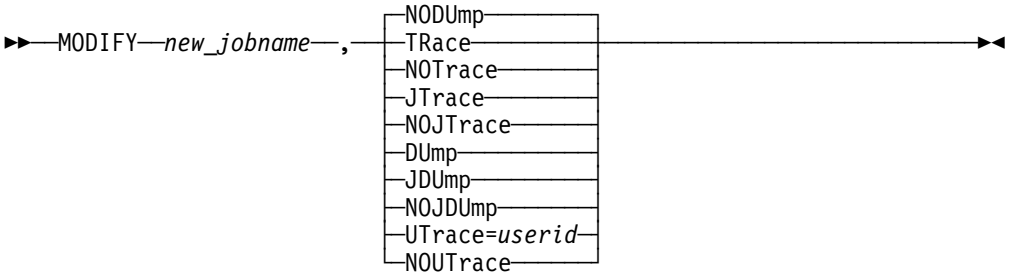
Tracing the OE FTP Server

Use the TRACE start parameter, or the TRACE statement in FTP.DATA, to start tracing during FTP initialization

Use the MODIFY command to start and stop tracing after initialization is complete:

Only FTP sessions established after trace is active can be traced. When tracing is stopped, sessions currently connected to the server will continue to be traced. New FTP sessions will not be traced.

Syntax



Parameters

TRACE

Enables a general trace for all clients connecting to the server.

NOTRACE

Disables the general trace.

JTRACE
Enables a JES trace for JES-related activity for all clients connecting to the server.

NOJTRACE
Disables the JES trace.

DUMP
Includes additional detailed activity in the log whenever the general trace is active.

NODUMP
Excludes detailed data from the general trace log. This is the default.

JDUMP
Includes additional detailed activity in the log whenever the JES trace is active.

NOJDUMP
Excludes detailed data from the JES trace log.

UTRACE=*userid*
Enables all levels of tracing for one specified user ID. All other trace options that were in effect are suspended.

NOUTRACE
Disables tracing for the specified user ID and resumes other traces that were in effect.

Examples
Enable JES tracing for an FTP server started with the FTPD procedure:

```
F FTPD1,JTRACE
```

where FTPD1 is the new jobname as given in EZYFT41I.

Related Topics
For more information on FTP server traces and how the parameter values interact, refer to *OS/390 TCP/IP OpenEdition Diagnosis Guide*.

Coexistence of Servers

1. It is possible to run both non-OE and OE FTP servers on the same system. The non-OE server can run only on the TCP/IP for MVS stack. The OE server can run on either the TCP/IP for MVS stack or the OS/390 TCP/IP OpenEdition stack.

   The simplest configuration running both servers would run the OE FTP server on OS/390 TCP/IP OE and the non-OE FTP server on TCP/IP for MVS. In this case both servers can use the same (default) port number. To tell OE to send the socket calls from the OE FTP server only to OS/390 TCP/IP OpenEdition, code the BPXPRMxx FILESYSTYPE statement to use INET as follows:

   ```
   FILESYSTYPE TYPE(INET) ENTRYPOINT(EZBPFINI)
   ```

   For further information, see scenario A of chapter 3 in the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

   If you choose to run both the OE FTP server and the non-OE FTP servers on the TCP/IP for MVS stack, you will have to:

- Enable the stack to differentiate between the two servers by using different port numbers for each of the FTP servers. For more information, see “How Can TCP/IP V3R2 Distinguish between OE Telnet and Non-OE Telnet?” in chapter 3 of the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.
- Code the BPXPRMxx FILESYSTYPE statement to use CINET, and include a SUBFILESYSTYPE for each of the TCPs which will support OE sockets.

  ```
  FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXTCINT)
  SUBFILESYSTYPE NAME(TCPIPA) TYPE(CINET) ENTRYPOINT(EZBPFINI)
     DEFAULT
  SUBFILESYSTYPE NAME(TCPIP32) TYPE(CINET) ENTRYPOINT(BPXTIINT)
  ```

  For more information, see scenario C in Chapter 3 of the *OS/390 TCP/IP OpenEdition Planning and Release Guide*.

2. Sharing of FTP.DATA files

   FTP.DATA data sets may be shared between different FTP servers with the following restriction:

   Unsupported statements. The OE FTP server will ignore any FTP.DATA statements that are valid for either the non-OE FTP server or the FTP client but are not specifically supported by the OE FTP server. ("Ignored" means that an informational message is issued indicating that the keyword is not supported and was ignored; processing continues with the next statement.)

   Similarly, the non-OE server and the FTP client will ignore statements that are valid only for the OE server. The client does not issue informational messages for ignored statements.

OE FTP Code Page Conversion

Code page conversion must be performed between EBCDIC and ASCII for:

- FTP subcommands and replies sent over the control connection
- Data transferred over the data connection

OE FTP uses *iconv* functions to establish ASCII-to-EBCDIC and EBCDIC-to-ASCII translate tables for the control connection and the data connection, with a default of 7-bit ASCII. In addition, OE FTP maintains support for the use of translate tables generated by the CONVXLAT utility for the control or data connection. You can use the OE FTP Server to retrieve data stored by the non-OE FTP servers, but you should use the same set of translate tables. To avoid data corruption due to code page differences, you should not use iconv to retrieve data stored by the non-OE FTP servers.

Once an end user has logged in, a 'site' subcommand can be used to change the ASCII being used on the control connection or the single byte translation for the data connection.

Code Page Conversions for the Control Connection

For the EBCDIC/ASCII conversions for the control connection, OE FTP uses either iconv or the existing support for single-byte translation tables.

Priority

The priority for establishing the EBCDIC/ASCII control connection is:

1. A new keyword in FTP.DATA to specify either 7-bit ASCII or an ASCII supported by iconv
2. Search order used by the non-OE server to locate a TCPXLBIN data set:
   a. original jobname.SRVRFTP.TCPXLBIN
   b. hlq.SRVRFTP.TCPXLBIN
   c. original jobname.STANDARD.TCPXLBIN
   d. hlq.STANDARD.TCPXLBIN
3. 7-bit ASCII
4. Internal (hard-coded) 7-bit tables

Code Page Conversions for the Data Connection

For the transfer of data on the data connection, OE FTP will support:

1. All single-byte conversions available through *iconv*. For example, Country-Extended_code-Pages (CECPs) <-> ISO8859-1 and IBM-1047 <-> IBM-850 conversions are available for data transfers.

   Note: The double-byte iconv convertors are not supported by OE FTP.
2. Translate tables that are generated by the CONVXLAT utility or the existing translate tables that are shipped with TCP/IP Version 3 Release 2. You may wish to use these tables to retrieve data that has been stored at MVS hosts. The translate tables that are shipped withTCP/IP Version 3 Release 2 are unique to TCP/IP for MVS. (Both single-byte and double-byte data conversions are supported with existing tables.)

Priority

The priority for establishing ASCII/EBCDIC conversions for the data connection is:

1. SYSFTSX DD statement in the startup proc, where the named data set contains CONVXLAT-generated translate tables. The data set can be an MVS data set or an HFS file.
2. SBDATACONN keyword in FTP.DATA
3. Search order used in V3R2:
   a. original jobname.SRVRFTP.TCPXLBIN
   b. hlq.SRVRFTP.TCPXLBIN
   c. original jobname.STANDARD.TCPXLBIN
   d. hlq.STANDARD.TCPXLBIN

   where the MVS data set contains CONVXLAT-generated translate tables
4. The same conversions established for the control connection

Chapter 8. Configuring the OE Remote Execution Server

Before You Configure...:

Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

This chapter describes how to configure and operate the OE Remote Execution server.

The Remote Execution Protocol Daemon (REXECD) is the server for the REXEC routine. REXECD provides remote execution facilities with authentication based on user names and passwords.

The Remote Shell Server (RSHD) is the server for the remote shell (RSH) client. The server provides remote execution facilities with authentication based on privileged port numbers, user IDs, and passwords.

Installation Information

This section describes the HFS files used by OE REXECD and OE RSHD.

HFS Files for OE REXECD

The HFS files used by OE REXECD and their locations in the HFS are as follows:

/etc/services
: The ports for each application are defined here.

/etc/syslog.conf
: The configuration parameters for usage of syslogd are defined in this file.

/etc/inetd.conf
: The configuration parameters for all applications started by inetd are defined in this file.

/usr/sbin/orexecd
: This is the server.

  If BPX.DAEMON is specified, then the sticky bit must be set on, and /usr/sbin/orexecd and orexecd must reside in an authorized MVS data set.

/usr/lib/nls/msg/C/rexdmsg.cat
: This is the message catalog used by the OE REXECD server.

  Where the server looks for the message catalog (rexdmsg.cat) depends on the value of NLSPATH and LANG environment variables. If you want to store the msg.cats elsewhere, you need to change the NLSPATH or the LANG environment variables. If rexdmsg.cat does not exist, the software will default to the messages hard-coded within the software. These messages duplicate the English message catalog that is shipped with the product.

HFS Files for OE RSHD

The HFS files used by OE RSHD and their locations in the HFS are as follows:

/etc/services
: The ports for each application are defined here.

/etc/syslog.conf
: The configuration parameters for usage of syslogd are defined in this file.

/etc/inetd.conf
: The configuration parameters for all applications started by inetd are defined in this file.

/usr/sbin/orshd
: This is the server.

  If BPX.DAEMON is specified, the sticky bit must be set on, and /usr/sbin/orshd and orshd must reside in an authorized MVS data set.

/usr/sbin/ruserok
: This is an optional user exit that will authenticate users loggin into the OE RSHD server with a null password. Parameters are passed to the exit in the following order:

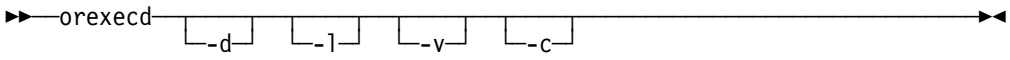
  1. hostname
  2. local user's UID
  3. remote userid
  4. local userid.

  The exit passes back a non-zero return code if authentication was unsuccessful.

/usr/lib/nls/msg/C/rshdmsg.cat
: The message catalog associated with the OE RSHD client is stored here. If this file does not exist, the software will default to the messages hard-coded within the software. These messages duplicate the English message catalog that is shipped with the product.

OE REXECD Command (orexecd)

Following is the syntax for the orexecd command:

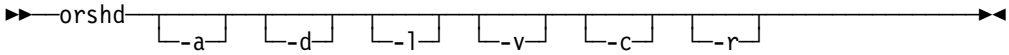


Following is a description of the supported options:

Option	Description
-d	Print debug information to syslogd.
-l	Write each successful login to syslogd with the remote user, remote system, local user, and the command executed.
-v	Write the title and ptf level to syslogd.
-c	Write all messages in uppercase.

OE RSHD Command (orshd)

Following is the syntax for the orexecd command:



Following is a description of the supported options:

Option	Description
-a	Look up hostname and check that the address and hostname correspond.
-d	Print debug information to syslogd.
-l	Write each successful login to syslogd with the remote user, remote system, local user, and the command executed.
-v	Write the title and ptf level to syslogd.
-c	Write all messages in uppercase.
-r	If a client passes a null password, invoke the /usr/sbin/ruserok user exit to authenticate the userid.

Chapter 9. Configuring Simple Network Management Protocol (SNMP) for OE

Before You Configure...:

> Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

This chapter describes how to configure the Simple Network Management Protocol (SNMP). The osnmp OMVS command is the new SNMP command used to access MIB object information.

Figure 3 illustrates the interface between TCP/IP and the implementation of SNMP.

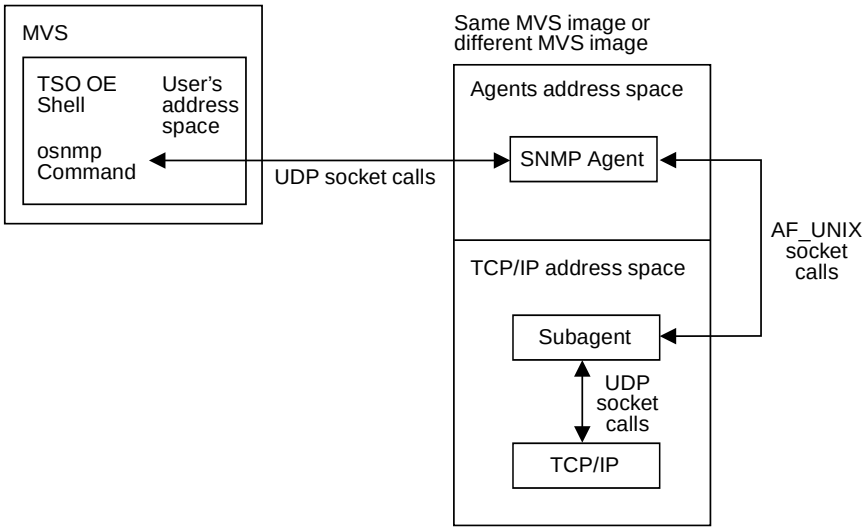


Figure 3. Overview of SNMP Support

This list illustrates the sequence of events from the time you issue an osnmp command until you receive the response:

1. The user or an OMVS shell script issues an osnmp command.
2. The osnmp command is validated by the osnmp command processor.
3. The command processor passes the request to an SNMP agent. The agent can be at another host (MVS, VM, OS/2, AIX, etc.).
4. The SNMP agent validates the request and, if necessary, sends it to an SNMP subagent. Requests for system group, saMIB, dpi, and snmp objects are handled by the agent and all others are handled by a subagent. To determine which objects are handled by the agent and which by a subagent, see the “Managment Information Base (MIB) Objects” appendix in the OS/390 TCP/IP OpenEdition User's Guide.
5. The agent sends the response to the originator of the request (i.e. the osnmp command). The osnmp command then displays the response.

The SNMP agent and the SNMP subagent both record trace information via the OMVS syslogd daemon using the *daemon* facility. Also, the osnmp command writes

some messages via the OMVS syslogd daemon. For detailed information regarding syslogd and specifying the daemon facility in the /etc/syslog.conf configuration file, see Appendix C, “Description of Syslog Daemon (syslogd)” on page 259.

Complete the following steps to configure SNMP:

1. Configure the SNMP agent (OSNMPD)
2. Configure the SNMP subagent
3. Configure the osnmp command
4. Configure the ATM Open Systems Adapter 2 (ATM OSA-2) support

Step 1: Configure the SNMP Agent (OSNMPD)

The OS/390 TCP/IP OpenEdition SNMP agent accepts both SNMPv1 and SNMPv2 (SNMPv2c only) requests.

To configure the SNMP agent, perform the following tasks:

- Provide TCP/IP Profile statements
- Provide Trap Destination information
- Provide Community Name information
- Provide MIB object Configuration information
- Starting the SNMP agent (OSNMPD)

Provide TCP/IP Profile Statements

There are two TCP/IP ports used by the SNMP agent, one for receiving incoming requests and one for sending traps to managers.

The default port used by the SNMP agent to receive incoming requests is 161. If you want the agent to use port 161 for this purpose and want to insure that no other application uses this port, you must specify the following PORT statement in your profile data set:

```
PORT
  161 UDP OSNMPD              ; SNMP Agent port for SNMP requests
```

If you want to define another port other than 161 for SNMP requests, you must do the following:

- Specify the port number to use on the SACONFIG profile statement
- Start the agent with a `-p` parameter.
- Make an entry in the snmpv2.conf file with the correct port number. For details on creating this entry, see the description for *targetAgent* in “/etc/snmpv2.conf Statement Syntax” on page 210.

The SNMP agent uses port 162 for sending traps to the managers specified in SNMPTRAP.DEST. To insure that no other applicaton uses this port, you must specify the following PORT statement in your profile data set:

```
PORT
  162 UDP OMVS                ; osnmp command port for receipt of traps
```

Provide Trap Destination information

Traps are unsolicited messages that are sent by an SNMP agent to an SNMP network management station. An SNMP trap contains information about a significant network event. The management application running at the management station interprets the trap information sent by the SNMP agent.

The following traps are generated by an SNMP agent in OS/390 TCP/IP OpenEdition:

- AUTHENTICATION_FAILURE
- COLD_START
- LINK_DOWN
- LINK_UP
- PVC creation
- PVC deletion

  PVC traps are reported for ATM Permanent Virtual Connections. For further information, see “Step 4: Configure the ATM Open Systems Adapter 2 (ATM OSA-2) Support” on page 214.

Note: The SNMP agent Distributed Program Interface allows external processes (which might be running on another host) to generate SNMP traps. This can allow for support of other types of traps. For more information about SNMP DPI, see the OS/390 TCP/IP OpenEdition Programmer's Reference.

To use traps, you must provide SNMPTRAP.DEST information defining a list of managers to which traps are sent. The SNMPTRAP.DEST information is optional. If no trap destination file is found, then the SNMP agent will send traps to the loopback IP address (127.0.0.1) and will issue a warning message indicating that defaults are in effect. If a trap destination file exists, but is empty, no traps will be sent.

SNMPTRAP.DEST Search Order

To access the SNMPTRAP.DEST information, the search order is:

- /etc/snmptrap.dest HFS file
- The data set specified on SNMPTRAP DD statement in the agent procedure
- '*jobname*.SNMPTRAP.DEST', where *jobname* is the name of the job used to start the SNMP agent
- 'SYS1.TCPPARMS(SNMPTRAP)'
- '*hlq*.SNMPTRAP.DEST', where *hlq* either defaults to TCPIP or is specified on the DATASETPREFIX statement in the TCPIP.DATA file being used.

If creating a data set, you can specify a sequential data set with the following attributes: RECFM=FB, LRECL=80, and BLKSZ=3120. Other data set attributes might also work, depending on your installation parameters.

SNMPTRAP.DEST Statement Syntax

- The SNMPTRAP.DEST statements list managers who are to receive the traps, and the protocol used to send traps. The format of a statement is:

  ```
  manager UDP
  ```

  The *manager* is the host to which the trap is to be sent. This can be a host name, or it can be the IP address of the host. If a host name is specified the

value may contain both uppercase and lowercase letters and is case-sensitive. The protocol must be UDP. There should be one entry in the data set for each host to which you want to send traps.

- All parameters for each host must be on the same statement.
- Sequence numbers are not allowed on the statements.
- Comments begin with either '*' or '#'.

SNMPTRAP.DEST Example

For example, the SNMPTRAP.DEST statements could be specified as follows:

```
# SNMP Trap Destination information
124.34.216.1 UDP
MVSSYS2      UDP
```

Provide Community Name information

SNMP agents are accessed by remote network management stations. To allow network management stations to send inquiries to the SNMP agent, you may provide PW.SRC information which defines a list of community names and IP addresses that can use these community names. The community name operates as a password when accessing objects on a destination SNMP agent.

The PW.SRC information is optional. If no PW.SRC information is found, then the SNMP agent will accept requests with a community name of 'public' from any IP address. If a PW.SRC file exists, but is empty, and if no community name is specified on the `-c` parameter at the agent invocation, then no requests will be accepted by the agent.

PW.SRC Search Order

To access the PW.SRC information, the search order is:

- /etc/pw.src HFS file
- The data set specified on SYSPWSRC DD statement in the agent procedure
- '*jobname*.PW.SRC', where *jobname* is the name of the job used to start the SNMP agent
- 'SYS1.TCPPARMS(PWSRC)'
- '*hlq*.PW.SRC', where *hlq* either defaults to TCPIP or is specified on the DATASETPREFIX statement in the TCPIP.DATA file being used.

If creating a data set, you can specify a sequential data set with the following attributes: RECFM=FB, LRECL=80, and BLKSZ=3120. Other data set attributes might also work, depending on your installation parameters.

PW.SRC Statement Syntax

- The PW.SRC statements specify community names and hosts that can use each community name. The format of a statement is:

  community_name desired_network snmp_mask

  The *community_name* can be up to 15 characters in length. This value can contain both uppercase and lowercase letters; however, it is case-sensitive. In any requests received by the SNMP agent, the *community_name* must match the *community_name* specified in PW.SRC exactly, including the correct case.

- All parameters for each community must be on the same statement.
- Sequence numbers are not allowed on the statements.
- Comments begin with either '*' or '#'.

PW.SRC Example

For example, the PW.SRC statements could be specified as follows:

```
passwd1 9.0.0.0      255.0.0.0
passwd2 129.34.81.22 255.255.255.255
```

The community name of an incoming SNMP request is compared to the known community names. If a match is found, then the IP address of the incoming request is logically ANDed with the *snmp_mask* of the PW.SRC statement. The result of the logical ANDing process is compared with the *desired_network*. If they match, the request is accepted.

In the preceding example, if a request for *community_name* `passwd1` is received from the IP address 9.34.22.122, IP address 9.34.22.122 is ANDed with 255.0.0.0. The result is 9.0.0.0, which equals the specified *desired_network* for `passwd1`, so this request is accepted. In `passwd2`, if the *community_names* match, only requests from host 129.34.81.22 are accepted.

If the *community_names* do not match, or the IP address ANDed with the *snmp_mask* does not match, an `AUTHENTICATION_FAILURE` trap is sent if both of the following are true:

- A destination entry exists in SNMPTRAP.DEST
- Authentication failure traps have been enabled. These traps are enabled by setting MIB object "snmpEnableAuthenTraps.0" to 1. For an explanation of setting this object, see “Provide MIB Object Configuration information.”

A *desired_network* and *snmp_mask* of all zeros allows anyone with the correct *community_name* to make requests.

Provide MIB Object Configuration information

An installation can set values for selected MIB objects by providing OSNMPD.DATA information. If no OSNMPD.DATA information is found, the defaults for these MIB objects are as follows:

Object	Default
dpiPathNameForUnixStream	The default is /tmp/dpi_socket.
sysDescr	SNMPv2c agent version 1.0 with DPI version 2.0 (15 May 1996). The maximum length of this object is 255 octets.
sysContact	Zero length string. The maximum length of this object is 255 octets.

sysLocation

Zero length string. The maximum length of this object is 255 octets.

sysName

If the environment variable HOSTNAME exists, its value is used. Otherwise, the default value identifies the OS/390 MVS system under which the agent is running. The maximum length of this object is 255 octets.

sysObjectId

1.3.6.1.4.1.2.3.13

sysServices

A single octet that defaults to 0. See MIB II description for this object.

snmpEnableAuthenTraps

Default value is 2 which means "off" (that is, traps are disabled).

saDefaultTimeout

5 seconds

saMaxTimeOut

600 seconds

saAllowDuplicateIDs

Default is 1, which means "yes" (that is, allow multiple instances of a subagent).

For information about where these MIB objects are defined, see the “Management Information Base (MIB) Objects” appendix in the OS/390 TCP/IP OpenEdition User's Guide.

OSNMPD.DATA Search Order

To access the OSNMPD.DATA information, the search order is:

- /etc/osnmpd.data HFS file
- The data set specified on the OSNMPD DD statement in the agent procedure
- '*jobname*.OSNMPD.DATA', where *jobname* is the name of the job used to start the SNMP agent
- 'SYS1.TCPPARMS(OSNMPD)'
- '*hlq*.OSNMPD.DATA', where *hlq* either defaults to TCPIP or is specified on the DATASETPREFIX statement in the TCPIP.DATA file being used.

If creating a data set, you can specify a sequential data set with the following attributes: RECFM=FB, LRECL=80, and BLKSZ=3120. Other data set attributes might also work, depending on your installation parameters.

OSNMPD.DATA Statement Syntax

- The OSNMPD.DATA statements specify MIB objects and their values. The format of each statement is:

  ```
  object_name   value
  ```

  The following rules apply:

  - There can only be one *object_name* and associated *value* per statement.
  - The *object_name* and associated *value* (if the *value* is a display or octet string) are case-sensitive and will be saved in mixed case.
  - Any display or octet string *value* which has imbedded white space must be enclosed in double quotes. For example, see sysName in the example below.
  - If the *value* is a display or octet string, it must be enclosed within double quotes.
- All parameters for each object must be on the same statement.
- Sequence numbers are not allowed on the statements.
- Comments begin with either '*' or '#'.

OSNMPD.DATA Example

A sample of OSNMPD.DATA is installed as HFS file /usr/lpp/tcpip/samples/osnmpd.data. This sample, shown below, can be modified for your installation.

```
#
# osnmpd.data sample
#
# Sample file for setting MIB variables and options for
# the SNMPv2 Agent provided by TCP/IP for MVS
#
#      Licensed Materials - Property of IBM
#      This product contains "Restricted Materials of IBM"
#      5645-001 5655-HAL (C) Copyright IBM Corp. 1996.
#      All rights reserved.
#      US Government Users Restricted Rights -
#      Use, duplication or disclosure restricted by
#      GSA ADP Schedule Contract with IBM Corp.
#      See IBM Copyright Instructions.
#
  sysDescr "SNMPv2c agent version 1.0 with DPI version 2.0"
  sysContact  "Unknown"
  sysLocation "Unknown"
  sysName "TCP/IP for MVS Stack"
  sysObjectID     "1.3.6.1.4.1.2.3.13"
  sysServices       0
  snmpEnableAuthenTraps 1
  saDefaultTimeout    6
  saMaxTimeout  700
  saAllowDuplicateIDs   2
  dpiPathNameForUnixStream "/tmp/dpi_socket"
```

Starting the SNMP Agent (OSNMPD)

OSNMPD parameters

The SNMP agent is a separate address space (OSNMPD) that executes load module EZASNMPD. OSNMPD can be started without parameters or you can add any of the parameters listed below.

When starting OSNMPD from MVS, add the parameters to the PARMS= keyword on the EXEC statement of the OSNMPD cataloged procedure When starting OSNMPD from OMVS, specify the desired parameters on the osnmpd command.

Note: The parameters are case sensitive. They must be entered in lower case.

Parameter	Description
-c *community*	A community name is a password that can accompany an SNMP request that the agent receives. Specifying a *community* on this parameter when starting the agent causes the *community* to effectively be added to the list of community names in PW.SRC, with a mask and IP address of zeros. So, any request received with this *community* would be authenticated, i.e. the request would be accepted from any IP address. The default community name is 'public'.
-d *level*	Specifies the level of tracing to be started. The valid values for *level* are 0 to 255. If the -d parameter is not specified then the default level of 0 is used, meaning no tracing will be done. If the -d parameter is specified without a *level*, then a level of 31 is used, meaning all SNMP requests/responses/traps and DPI activity will be traced. There are 8 levels of tracing provided. Each level selected has a corresponding number. The sum of the numbers associated with each level of tracing selected is the value which should be specified as *level*. Once the agent is started, tracing options can be dynamically changed using the MVS MODIFY command. For more information on agent tracing, see the OS/390 TCP/IP OpenEdition Diagnosis Guide. The numbers for the trace levels are: **0** No tracing (default) **1** Trace SNMP requests **2** Trace SNMP responses **4** Trace SNMP traps **8** Trace DPI packets **16** Trace DPI internals (currently, no specific traces are recorded for this trace level) **32** Agent internal trace **64** Agent internal trace plus extended storage dump traces **128** Agent internal trace plus extended storage dump traces plus additional information
-p *port*	listen for SNMP packets on this port. The default is port 161. If you change the port to something other than 161, you must also specify the new port on the SACONFIG statement.
-s *socketname*	The name of the UNIX socket to be used in accepting requests from subagents that communicate with the agent via AF_UNIX connections. This value can be configured either by specifying it on the `-s` parameter or by specifying it as the value of the `dpiPathNameForUnixStream` MIB object in

OSNMPD.DATA. The default is `/tmp/dpi_socket`. The file permission bits for this file must be read and write for the subagents to connect.

?

Display the usage statement for the command. If this parameter is specified, all other parameters are ignored. If OSNMPD was started from MVS then the usage information is written to syslogd. If OSNMPD was started from OMVS then the usage information is displayed to the invoker of the command.

Sample JCL Procedure for Starting OSNMPD from MVS

Update cataloged procedure OSNMPD by copying the sample in *hlq*.SEZAINST(OSNMPDPR) to your system or recognized PROCLIB. Change the data set names as required to suit your local configuration. The OSNMPD address space requires access to the IBM C/370 Library during execution.

Parameters may be passed to the agent on the `PARMS=` keyword on the EXEC statement of the OSNMPD cataloged procedure. For a detailed explanation of the parameters, see “OSNMPD parameters” on page 205. Any agent parameters you wish to specify may be added as shown in the following example:

```
//OSNMPD EXEC PGM=EZASNMPD,REGION=4096K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON)/ -c abc -d 255 -p 761'
```

In this example, the agent will use port 761 to accept requests, community name 'abc' will be added to the list of commnity names supported by the agent, and all agent traces will be activated. For more information on tracing, see the OS/390 TCP/IP OpenEdition Diagnosis Guide.

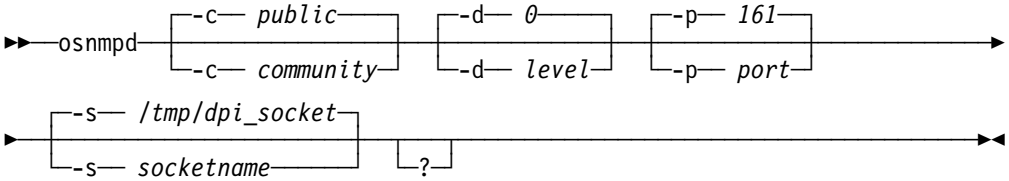
Shown below is the sample OSNMPD procedure:

```
//OSNMPD    PROC
//*
//* Sample procedure for running the OE SNMP agent
//*
//* TCP/IP for MVS
//* SMP/E Distribution Name: AEZASMP1
//*
//*       Licensed Materials - Property of IBM
//*       This product contains "Restricted Materials of IBM"
//*       5645-001 5655-HAL (C) Copyright IBM Corp. 1996.
//*       All rights reserved.
//*       US Government Users Restricted Rights -
//*       Use, duplication or disclosure restricted by
//*       GSA ADP Schedule Contract with IBM Corp.
//*       See IBM Copyright Instructions.
//*
//OSNMPD EXEC PGM=EZASNMPD,REGION=4096K,TIME=NOLIMIT,
// PARM='POSIX(ON) ALL31(ON)/'
//*
//*   The C runtime libraries should be in the system's link list
//*   or this sample procedure will need to STEPLIB to them.
//*
//*   TCP/IP runtime libraries should also be in the system's link
```

```
//*   list.
//*
//*   OSNMPD must find the name (TCPIPJOBNAME in TCPIP.DATA) that
//*   it should be associated with. The OE function __iptcpn() is
//*   used to find this name. It is suggested that the parmlist
//*   be modified to set the environment variable
//*   RESOLVER_CONFIG to point to the correct resolver file when
//*   multiple INET Physical File Systems are started:
//*
//*   //      PARM=('POSIX(ON) ALL31(ON)',
//*   //      'ENVAR("RESOLVER_CONFIG=/etc/tcpv33a.data")/')
//*
//*   If only one INET PFS will be started then it is
//*   recommended that /etc/resolv.conf be used.
//*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN    DD DUMMY
//SYSMDUMP DD SYSOUT=*
//SYSERR   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//CEEDUMP  DD SYSOUT=*
```

Command for Starting OSNMPD from OMVS

Shown below is the command syntax used to start OSNMPD from OMVS.



Note: Each parameter must be separated by a blank. Parameter values can be specified in mixed case.

To run the SNMP agent in background, you must add an ampersand (&) to the command and the issuer of the command must be in OMVS superuser mode. For a detailed explanation of the osnmpd parameters, see “OSNMPD parameters” on page 205.

Any agent parameters you wish to specify may be added as shown in the following example:

```
osnmpd -c abc -d 255 -p 761
```

In this example, the agent will use port 761 to accept requests, community name 'abc' will be added to the list of commnity names supported by the agent, and all agent traces will be activated. For more information on tracing, see OS/390 TCP/IP OpenEdition Diagnosis Guide.

Step 2: Configure the SNMP Subagent

There are two statements in the profile data set used to configure the SNMP subagent, the SACONFIG and ITRACE statements.

- SACONFIG

  Use the SACONFIG statement to configure the subagent. The SACONFIG parameters determine whether or not the subagent is automatically started at TCP/IP initialization, what port number to use to contact the agent, and other configuration values. For a detailed explanation of this statement, see "SACONFIG Statement" on page 89.

- ITRACE

  Use the ITRACE statement to determine what trace information, if any, should be recorded by the subagent. For a detailed explanation of this statement, see "ITRACE Statement" on page 76.

Step 3: Configure the osnmp command

The osnmp command is used to send SNMP requests to SNMP agents on either local or remote hosts. The requests can be either SNMPv1 or SNMPv2. For SNMPv2 requests, the /etc/snmpv2.conf configuration HFS file is required. The *winSnmpName* specified on a /etc/snmpv2.conf statement can be used as the value of the -h parameter on the osnmp command. For a detailed explanation of the parameters you can specify on the osnmp command, see the OS/390 TCP/IP OpenEdition User's Guide.

To configure the osnmp command, perform the following tasks:

- Provide osnmp SNMPv2 Configuration information
- Provide Community Name information
- Provide User MIB object information

Provide osnmp SNMPv2 Configuration information

The name of the configuration file must be snmpv2.conf and it must reside in the /etc directory (that is, /etc/snmpv2.conf). A sample of this file is installed as HFS file /usr/lpp/tcpip/samples/snmpv2.conf. This sample, shown below, should be copied over to the /etc directory and modified for your installation.

```
# snmpv2.conf
#
# Sample file showing format of configuration file for the osnmp
# command.
#
#      Licensed Materials - Property of IBM
#      This product contains "Restricted Materials of IBM"
#      5645-001 5655-HAL (C) Copyright IBM Corp. 1996.
#      All rights reserved.
#      US Government Users Restricted Rights -
#      Use, duplication or disclosure restricted by
#      GSA ADP Schedule Contract with IBM Corp.
#      See IBM Copyright Instructions.
#
#
mvs1  1.23.456.78  snmpv2c
mvs2       mvs2c   snmpv2c
mvs3  mvs3:1061    snmpv2c
router1_v2c         router1 snmpv2c
router1_v2u_noauth  router1 snmpv2u operator password - none -   - - -
router1_v2u_auth    router1 snmpv2u operator password - auth MD5 - - -
```

/etc/snmpv2.conf Statement Syntax

The configuration file is used when sending requests to the SNMPv2 nodes in your network. The types of SNMPv2 nodes currently supported are:

snmpv2c IESG Draft-Standard "Community Based SNMPv2" (SNMPv2C) Agents.
snmpv2u USEC, "User-Based Security Model" Secure SNMPv2 Agents.

Note: The OS/390 TCP/IP OpenEdition SNMP agent does not support snmpv2u requests. Some of the fields on the statements in the configuration file are not used by the SNMP agent. These include userName, password, context, Qos, authProto, authKey, privProto, and privKey.

The following rules apply to the statements:

- The syntax of a statement in the configuration file is:

```
winSnmpName targetAgent v2Admin userName password context Qos authProto authKey privProto privKey
```

Where:

winSnmpName The name by which osnmp functions can locate an entry in this configuration file. Specified on the `-h` option. (Max 32 characters.)

targetAgent Hostname or IP address of the node of the target agent (Max 80 characters). To direct the command to a port other than 161, specify *host:port#*. For example, for port 222 at mvs150, specify mvs150:222.

v2Admin Specifies the SNMPv2 administration model supported by this node. Valid values are...

- snmpv2c - Community Based SNMPV2
- snmpv2u - USEC Based Secure SNMPV2

userName The SNMPv2 USEC username which has access to this node. Valid only for with the snmpv2u admin model. (Minimum 1 character, Maximum 16 characters)

password The SNMPv2 USEC password for "userName" above. Valid only for use with the snmpv2u admin model. If a password is specified, it will be used to automatically generate any needed keys and the "authKey" and "privKey" fields below will be IGNORED. If no password desired, set field to a single dash - . (Minimum 8 characters, Maximum 64 characters.),

context The SNMPv2 USEC context selector for this node. Valid only for with the snmpv2u admin model. If the blank "" context selector is desired, set this field to a single dash - . (Maximum 40 characters).

Qos The SNMPv2 USEC Quality Of Service wanted. Specify one of the following values:

none	no Authentication, no Privacy
auth	Authentication but no Privacy
priv	Authentication and Privacy

authProto SNMPv2 USEC Authentication Protocol to be used. Currently only one value can be specified: MD5 If no authentication is used, set field to a single dash - .

authKey SNMPv2 USEC Authentication Key to be used when sending an authenticated request to an snmpv2u agent. If the "password" field above is specified, it will be used to automatically generate the key and this field will be IGNORED. Valid only for the snmpv2u admin model. If no key is desired, set field to a single dash - . (32 hex digits)

privProto SNMPv2 USEC Privacy Protocol to be used. Only the

```
DES
```

value can be specified. If no privacy is used, set field to a single dash - .

privKey SNMPv2 USEC Privacy Key to be used when sending an encrypted request to an snmpv2u agent. If the "password" field above is specified, it will be used to automatically generate the key and this field will be IGNORED. Valid only for the snmpv2u admin model. If no key is desired, set field to a single dash - . (32 hex digits)

- All parameters for each *winSnmpName* must be on the same statement.
- A “-” indicates a default value for the parameter in that position.
- Sequence numbers are not allowed on the statements.
- Comments begin with '#'.

Examples

- Example 1:

  The following entry defines an SNMPv2c node to WinSNMP...

  ```
  router1 router2 snmpv2c
  ```

  where router1 is the winsnmp name used with the —h parameter and router2 is the host name for the snmpv2c agent.

- Example 2:

  The following defines a Secure SNMPv2 USEC node to WinSNMP... Communication with that node will use authenticated SNMP messages.

```
router1 router1 snmpv2u operator - - auth MD5 000102030405060708090a0b0c0d0e0f - -
```

- Example 3:

  Secure SNMPv2 USEC node using Password instead of keys.

```
router1 router1 snmpv2u operator password - auth MD5 - - - -
```

- Example 4:

  Use different protocol (SNMPv2C or SNMPv2U) and use different Quality of Service for SNMPv2U

```
router1_v2c         router1 snmpv2c
router1_v2u_noauth  router1 snmpv2u operator password - none -   - - -
router1_v2u_auth    router1 snmpv2u operator password - auth MD5 - - -
```

- Example 5:

  The following defines port 1061 as the port used to send SNMP requests to an SNMP agent.

```
mvs3 mvs3:1061  snmpv2c
```

Provide Community Name information

Community name information for the osnmp command can be provided in an HFS file named /etc/pw.src.cli. The information in this file is the same as that of the agent's PW.SRC information and the statements have the same syntax. This file is used to validate the community name on SNMP traps and responses from an SNMP agent.

Since the same community name that is sent on the command is used by the agent to return a response, the community name must be valid for both the IP address from which the request originates and for the IP address of the agent. If you want the osnmp command to accept responses and traps from a different set of IP addresses other than the set of IP addresses from which the agent will accept requests, you must use the /etc/pw.src.cli file to define a different desired_network and snmp_mask for that community name.

If this file does not exist, the osnmp command will follow the PW.SRC search order (defined earlier for the agent) to obtain the community name information. For a description of this search order and of the statement syntax for this file, see “Provide Community Name information” on page 202.

Provide User MIB object information

If you want to use MIB objects which are not defined in any compiled MIB, then you can define them in an /etc/mibs.data HFS file. For a list of object in the compiled MIB, see the “Management Information Base (MIB) Objects” appendix in the *OS/390 TCP/IP OpenEdition User's Guide.* A sample of the /etc/mibs.data file is installed as HFS file /usr/lpp/tcpip/samples/mibs.data. This sample, shown below, should be copied over to the /etc directory and modified for your installation.

```
#      Licensed Materials - Property of IBM
#      This product contains "Restricted Materials of IBM"
#      5645-001 5655-HAL (C) Copyright IBM Corp. 1996.
#      All rights reserved.
#      US Government Users Restricted Rights -
#      Use, duplication or disclosure restricted by
#      GSA ADP Schedule Contract with IBM Corp.
#      See IBM Copyright Instructions.
# short name            OID                   type
#-------------------------------------------------

# my system  objects

myDescr              1.3.6.1.2.1.1.1.          display
myObjectid           1.3.6.1.2.1.1.2.          objectidentifier
myUptime             1.3.6.1.2.1.1.3.          Timeticks
myContact            1.3.6.1.2.1.1.4.          display
myName               1.3.6.1.2.1.1.5.          display
myLocation           1.3.6.1.2.1.1.6.          display
myServices           1.3.6.1.2.1.1.7.          integer

# DPI SAMPLE MIB:

dpiSimpleInteger    1.3.6.1.4.1.2.2.1.5.1.0   integer
dpiSimpleString     1.3.6.1.4.1.2.2.1.5.2.0   display
dpiSimpleCounter32  1.3.6.1.4.1.2.2.1.5.3.0   counter32
dpiSimpleCounter64  1.3.6.1.4.1.2.2.1.5.4.0   counter64
```

/etc/mibs.data Statement Syntax

The /etc/mibs.data statements can be used to specify character (usually called 'textual') names for MIB objects not defined in any compiled MIB supplied with OS/390 TCP/IP OpenEdition. You can then use these character/textual names as the name of the objects on the osnmp command.

- The maximum length of each statement in this file is 2048 bytes.
- The format of a statement in this file is:

  character_object_name object_identifier object_type

  Where:

  character_object_name The character/textual name of the MIB object. The *character_object_name* value can contain both uppercase and lowercase letters; however, it is NOT case-sensitive.

  object_identifier The ASN.1 value for the MIB object.

  object_type The SMI_type for the MIB object. The valid SMI_type values are:

  - bitstring
  - counter
  - counter32
  - counter64
  - display or display string
  - integer
  - integer32
  - ipaddress
  - gauge
  - gauge32

  - nsapaddress
  - null
  - objectidentifier or OID
  - octetstring
  - opaque
  - opaqueascii
  - timeticks
  - uinteger
- All parameters for each character/textual name must be on the same statement.
- Sequence numbers are not allowed on the statements.
- Comments begin with '#'.

Step 4: Configure the ATM Open Systems Adapter 2 (ATM OSA-2) Support

OS/390 TCP/IP OpenEdition does not support an ATM Port as a transport facility, but the SNMP subagent can interface with an ATM OSA-2 adapter via the Open Systems Adapter Support Facility (OSA/SF) for support of ATM Management.

Current support consists of:

- Data retrieval enablement for operational and performance management.
- Asynchronous SNMP Trap generation for operational management:
  - ATM Port enabled - LinkUp Trap
  - ATM Port disabled - LinkDown Trap
  - Permanent Virtual Circuit (PVC) creation - ibmMvsOsasfAtmPvcCreate Trap
  - Permanent Virtual Circuit (PVC) deletion - ibmMvsOsasfAtmPvcDelete Trap
- Provide method for assigning an IP Address to the ATM Port.

The following sections describe this support:

- OSA/SF Prerequisites
- Required TCP/IP Profile statements
- Multiple TCP/IP Instances Considerations

OSA/SF Prerequisites

The SNMP subagent provided by TCP/IP will connect to OSA/SF in order to provide for ATM Management. In order for a subagent to establish a connection to OSA/SF two OSA/SF components must be started:

- IOAOSASF

  IOAOSASF is a sample JCL procedure that can be used to start the main OSA/SF address space. The sample has a jobname of OSASF1.

- IOASNMP

  IOASNMP is a sample JCL procedure that starts the OSA/SF-provided OE transport application that interconnects a subagent with OSASF1.

These sample procedures and all entities that they call are provided with OSA/SF. For a detailed explanation of how to setup OSA/SF on your MVS system, see *Planning for the System/390 Open Systems Adapter Feature* and the *OS/390 Open Systems Adapter Support Facility User's Guide*. The primary purpose of OSA/SF is to manage OSA Adapters. It has been extended to support ATM Management via SNMP. An instance of IOAOSASF, IOASNMP, TCPIP and its subagent, and an SNMP agent must be started on every MVS image where ATM Management support is needed.

The recommended startup order is:

1. Start IOAOSASF and wait until it completely initializes. IOAOSASF must be started before IOASNMP.
2. Start TCP/IP and wait until TCP/IP completes its initialization. Actually IOAOSASF can be started after TCP/IP but must be started prior to the next step.
3. Start IOASNMP after TCP/IP is initialized. If starting multiple TCP/IP instances that run under OMVS as AF_INET Physical File Systems, wait until at least one TCP/IP where OSA/SF support was requested has initialized. OSA/SF support is requested by specifying the `OSASF` parameter on the `SACONFIG` statement in the profile data set for a TCP/IP instance. For a detailed description of the SACONFIG statement, see "SACONFIG Statement" on page 89.
4. Start the SNMP Agent, OSNMPD, for each TCP/IP instance where ATM Management support is desired.

On an MVS image only a single instance of either IOAOSASF or IOASNMP can (or should) be started. An attempt to start multiple copies of IOAOSASF will be rejected. Starting multiple copies of IOASNMP will yield unpredictable results.

Insure that OSA/SF is at Version 1 Release 2 level or higher with the OSA/SF APAR OW24712 applied.

Required TCP/IP Profile statements

For a detailed description of the statements mentioned here, see Chapter 3, "Configuring the TCPIP Address Space" on page 33. The following TCP/IP profile statements must be updated for ATM Management support:

- SACONFIG

  On the SACONFIG statement, ATM Management support must be enabled by specifying ATMENABLED. Omission of ATMENABLED when TCP/IP is started will result in no ATM Management support. The SACONFIG statement controls the operation of the subagent that runs in a TCP/IP address space as a separate task.

  The OSASF parameter specifies which port IOASNMP should use to Listen for connections from subagents to OSA/SF. For an explanation of the usage of this parameter when starting multiple TCP/IP instances, see "Multiple TCP/IP Instances Considerations" on page 217. It is recommended that the OSASF port be reserved by also specifying it on a PORT statement.

  For example:

  ```
  SACONFIG ATMENABLED OSASF 721
  ```

- PORT

Prereserve the port to be used in communication with OSA/SF.

For example:

```
PORT
   721  IOASNMP ; OSA/SF TCP/IP Communications
```

In the above example since IOASNMP runs as an OE application the port could have been reserved for OMVS. Review the /etc/services HFS file to insure that there are no port conflicts.

- DEVICE and LINK

  Provide DEVICE and LINK statements for any ATM Port for which you want SNMP ATM Management support. For example:

  ```
  DEVICE osaName ATM
  LINK portName ATM osaName
  ```

  The *osaName* must be the OSA name known to OSA/SF and the *portName* must be the port name known to OSA/SF.

  Note: At this time TCP/IP does not support an ATM Port as a transport facility. It should be defined to TCP/IP only if ATM Management support is needed. An attempt to start the ATM by *osaName* will fail.

- HOME

  The HOME statement can be used to assign an IP address to an ATM Port. The SNMP subagent retrieves the IP address specified on the HOME statement and requests OSA/SF to set this address in the ATM Port. Hosts with access to the ATM Port via its ATM network can then retrieve this IP address.

  You should pick an IP address that is defined to this TCP/IP instance as one of its local IP addresses, and which is reachable from whichever hosts can query the IP address directly from the ATM Port via its ATM network. The IP address is intended for use by these other hosts in issuing SNMP commands to the agent/subagent running under the TCP/IP instance where the HOME statement is specified.

  OSA/SF has the restriction that in a single OS/390 system only one OSA/SF can actually customize an OSA Adapter. But all instances of OSA/SF running in their own MVS image can retrieve data. The effect of this with respect to ATM Management is that only the subagent running in the same MVS image as the customizing OSA/SF can set the IP address stored in an ATM Port on the OSA Adapter. Therefore, you should only specify a HOME statement for an ATM port in the profile data set of the TCP/IP running in the same MVS image as the customizing OSA/SF.

  For example:

  ```
  HOME   9.67.1.2  portName
  ```

Remember when using the HOME and PORT statements that the first time they appear when processing a profile data set, they completely replace the existing list. Each subsequent use of these statements within the same profile data set acts as an addition to the existing HOME and PORT list.

Multiple TCP/IP Instances Considerations

Subagent connection to OSA/SF

When multiple OS/390 TCP/IP OpenEdition instances are active in the same MVS image, only one of the TCP/IP instances is connected to IOASNMP. In order for a TCP/IP instance to connect to IOASNMP the OSASF parameter must be specified on the SACONFIG Profile statement.

IOASNMP connects to a TCP/IP instance and acts as a server, receiving connections from those SNMP subagents where ATMENABLED was specified on the SACONFIG Profile statement. The result is that all these subagents connect through the same TCP/IP to IOASNMP in order to retrieve ATM information from OSA/SF. For a depiction of this process, see Figure 4.

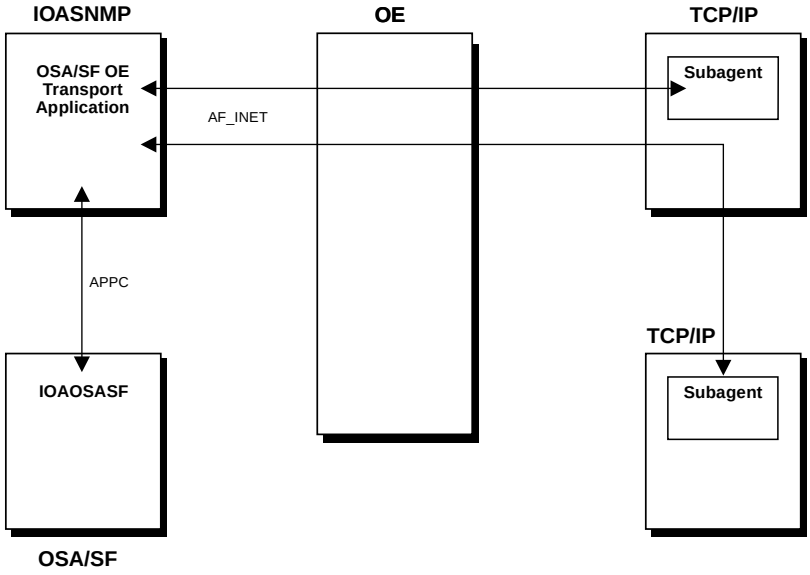


Figure 4. Subagent Connection to OSA/SF

If IOASNMP loses its connection to TCP/IP it terminates and needs to be restarted.

If the currently connected TCP/IP terminates, IOASNMP will attempt to connect to another TCP/IP instance for which the OSASF parameter was specified on the SACONFIG Profile statement, using the port number specified for the OSASF parameter. The subagents will also attempt to reconnect to OSA/SF via IOASNMP using this same OSASF port number. For this reason it is recommended that the same OSASF port number be used on the SACONFIG statement of every TCP/IP instance where the OSASF parameter is specified.

Whenever a socket error occurs on the OSA/SF socket, the connected subagents will issue the following message:

```
EZZ3219I SNMP SUBAGENT: DISCONNECTED FROM OSA/SF
```

When the subagent connection is reestablished, the following message is issued:

```
EZZ3218I SNMP SUBAGENT: CONNECTED TO OSA/SF
```

IP Address on HOME statement

When multiple OS/390 TCP/IP OpenEdition instances are active in the same MVS image, a HOME statement specifying an IP address for a particular ATM port should only be included in the profile data set of one of the TCP/IP instances if the IP addresses are different from each other.

The subagents in the TCP/IP instances will each send the IP address from their HOME statement to OSA/SF to be stored in the same ATM port, if the following are true:

- A HOME statement for the same ATM port is included in the profile data set of more than one TCP/IP instance.
- ATMENABLED was specified on the SACONFIG statement for the TCP/IP instances.

Since the subagents periodically refresh the MIB object values for their TCP/IP, the subagents repeatedly send the IP address from the HOME statements to OSA/SF to be stored in the ATM port. If the HOME statements specify different IP addresses for the same ATM port, then you will not be able to determine which IP address will be retrieved by a host directly from the ATM port via its ATM network.

Chapter 10. Configuring the OE RouteD Server

Before You Configure...:

> Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

This chapter describes how to configure the OE RouteD server. It explains OE RouteD's use of the Routing Information Protocol to help you decide if this server is suitable for your network. It also explains Virtual IP Addressing (VIPA) of OE RouteD which gives you the ability to include virtual routes in routing information. With virtual routes, routers in the network can route around interface, device and network failures. Examples for various configurations are given.

Understanding OE RouteD

The route daemon is a server that implements the Routing Information Protocol (RIP) (RFC 1058). It provides an alternative to the static TCP/IP gateway definitions. When configured properly the OS/390 server running with OE RouteD becomes an active RIP router in a TCP/IP network. The OE RouteD server dynamically creates and maintains the network routing tables using the Routing Information Protocol (RIP). The RIP protocol allows gateways and routers to periodically broadcast their routing tables to adjacent nodes. This enables the OE RouteD server to update the host routing tables. For example, the OE RouteD server can determine if a new route has been created, if a route is temporarily unavailable, or if a more efficient route exists. OE RouteD for OS/390 TCP/IP OpenEdition is functionally equivalent to RouteD for TCP/IP V3R2 with the following exceptions:

- OE RouteD is an OpenEdition (OE) application. It requires the Hierarchical File System (HFS) to run.
- OE RouteD can be started from an MVS procedure or it can be started from the OE shell command line.
- The messaging routines have changed. OE RouteD uses a standard message catalog. The message catalog must be in the HFS. The directory location for the message catalog path is set by the environment variables NLSPATH and LANG.
- The location and search order for configuration files are different.
- All messages and trace information is sent to the syslogd, except for output from the *-d* and *-dp* parameters, which is sent to STDOUT.
- The default mode of operation is for the program to close STDIN, STDOUT and STDERR. This allows for userids to cleanly exit the OE shell after starting the program in the background. As a consequence, the program printf statements are also disabled. A new parameter, "-ep", has been added to enable printf's. If this parameter is specified, the program should not be run in the background, because the userid will not be able to exit the shell until the background job has been killed.
- The command line options *-ep* and *-d* are new with this release.

- PARMS option *-k* has been added to the MODIFY command. For more information, see “Controlling OE RouteD with the MODIFY Command” on page 242.

Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is an Interior Gateway Protocol (IGP) designed to manage a relatively small network. IGPs are used to manage the routing information of a single autonomous system, or a single piece of the TCP/IP network. RIP has many limitations and is not suited for every TCP/IP environment. Before installing the OE RouteD server, read RFC 1058 to decide if RIP can be used to manage the routing tables of your network. See Appendix B, “Related Protocol Specifications” on page 253 for more information about RFC 1058.

RIP uses the number of hops, or *hop count*, to determine the best possible route to a host or network. The term *hop count* is also referred to as the *metric*. A gateway is defined as zero hops from directly connected networks, one hop from networks that can be reached through one gateway, and so on. In RIP, a hop count of 16 means infinity, or that the destination cannot be reached. This limits the longest path in the network that can be managed by RIP to 15 gateways.

The OE RouteD server broadcasts routing information to the gateway’s directly connected networks every 30 seconds. The server receives updates from neighboring gateways periodically and uses this information to update the routing tables. If an update has not been received from a gateway in 180 seconds (3 minutes), OE RouteD assumes the gateway is down and sets all the routes through that gateway to a metric of 16 (infinity). If an update has not been received from a gateway in another 120 seconds (2 minutes), OE RouteD deletes all of the routes through that gateway.

During the intervals specified by the*interface.scan.interval* and *interface.poll.interval* values on the OPTIONS statement, OE RouteD checks to determine if a local interface is up or down by scanning the TCP/IP interface tables. It also checks to see if an interface has been added or reactivated.

For networks that are not point-to-point, such as Token-Ring and Ethernet, OE RouteD receives its own broadcasted packets over the interfaces, provided that the interfaces are active. Other networks, such as point-to-point links, cannot be managed by OE RouteD unless a routed server is running on the host on the other end of the link. If the other host is not running routed, the OE RouteD server does not receive updates over the link and deletes all the routes over the point-to-point link. For more information, see “OE Routed Parameters” on page 234.

OE RouteD requires routers that do not support broadcasting (for example, CTC) to be active gateways since OE RouteD uses link-level broadcasting to send routing updates. For more information on how to manage CTC networks, see “OE RouteD Active Gateways” on page 225. The OE RouteD server never sends routing updates to the CTC network, because OE RouteD uses link-level broadcasting to send routing updates.

RIP assumes that the entire autonomous system has consistent subnets and uses a single subnet mask. The subnet mask for a subnet is not passed in the routing information; it is assumed that the gateway receiving the information is using the same subnet mask as the gateway sending the information. If different subnet masks are used throughout the network, you should not use RIP.

Note: Routing Information Protocol (RIP) uses the term *gateway* rather than the more correct term *router*. The term *gateway* is used when describing RIP and OE RouteD, to be compatible with RFC 1058. Only RIP Version 1 is currently supported by OE RouteD.

Primary and Alternate Network Attachments

Multiple attachments and IP addresses on the same LAN are supported, providing redundant paths to other hosts or routers on directly attached LANs. When multiple attachments to the same LAN are configured, one attachment is the primary path to hosts and routers on that LAN and others are secondary.

The primary path to hosts and routers on the LAN is defined in one of two ways:

- The first primary path can be specified in the PRIMARYINTERFACE statement. On remaining attachments, the first link for each defined network or subnetwork in the list of HOME addresses is primary.
- If no PRIMARYINTERFACE statement is configured, the first link for each defined network or subnetwork in the list of HOME addresses is primary.

All other interfaces on directly connected LANs are secondary.

MVS TCP/IP uses the primary interface for all outbound traffic to hosts and routers on directly attached LANs, as long as the primary interface is functioning. If a primary interface fails, outbound traffic is sent on a secondary interface.

Virtual IP Addressing (VIPA)

The purpose of Virtual IP Addressing (VIPA) is to free other TCP/IP hosts from dependence on particular network attachments to MVS. Prior to VIPA, other hosts got bound to one of MVS TCP/IP's home IP addresses and, therefore, to a particular network attachment (for example, a controller or adapter) to MVS. VIPA provides an IP address that selects a TCP/IP image (and MVS system if there is only one image on an MVS system) without selecting a specific network attachment. Other hosts that connect to MVS TCP/IP applications can send data to an MVS VIPA via whatever paths are selected by the routing protocols. VIPA provides tolerance of failures of MVS network attachment hardware.

VIPA uses a virtual device and a virtual IP address. The virtual device will always be active and never see a failure. A virtual IP address will be the home address for the virtual device, but there will be no physical interface associated with it. Inbound packets that have the virtual IP address as the destination can be routed through any one of the real physical interfaces to MVS. Failure of a real MVS network interface is handled by routing inbound traffic to another interface.

The OE RouteD application running on the OS/390 server provides RIP support. OE RouteD includes the network or subnetwork address of the virtual IP address in the routing broadcast information. Adjacent routers learn the virtual IP address from the broadcasts and can use it to reach the destination at the OS/390 server. OE RouteD provides the following functions:

- Automatic and transparent recovery from controller failure.

  When a controller (for example, 3172, 3745, or channel-attached RS/6000) fails, if there is another controller that provides alternate paths to the destina-

tion, and if other hosts are connecting with MVS TCP/IP applications via MVS virtual IP addresses:

  - MVS TCP/IP will detect the failure, find an alternate path for each LAN, and route outbound traffic to hosts and routers on those LANs via the alternate paths.
  - OE RouteD will advertise routes to VIPA(s) to adjacent routers on all appropriate links (including LANs, CTC, and IP/CDLC) using the alternate paths.
  - Routed running on adjacent routers and hosts will register these advertised paths to the VIPAs and reroute inbound traffic destined for VIPAs through the alternate connections.
  - The result is fault tolerance for both inbound and outbound MVS TCP/IP traffic without need to re-establish the active TCP connections that were using the failed link.

- Automatic and transparent recovery from interface failure.

  Assume that multiple network interface adapters (for example, an Ethernet and a Token Ring) are installed on a controller. In this configuration, the interfaces are configured as primary and backup. OE RouteD will add a route to one of the interfaces to reach the destination. When the interface containing the destination route fails, OE RouteD has the ability to switch outbound traffic to an available backup. OE RouteD will advertise routes to the VIPA(s) over the backup paths. Backup interfaces and VIPAs provide for redundant connections to be used in case of interface failure.

  - Automatic and transparent recovery from network failures (where the network has the necessary redundancy).
  - OE RouteD detects when network failures have occurred and dynamically switches traffic to alternate routes.

- Recovery from OS/390 server failure (where an alternate host has the necessary redundancy).

  Assume that an alternate OS/390 server is installed to serve as a backup and VIPA is configured on the primary host. In case of a primary host failure, the backup host can be reconfigured to use the primary host's VIPA. Client/server sessions on the primary host will be disrupted but they can be reestablished on the backup host using the primary host's VIPA as a destination address. The backup host with the reassigned VIPA can be configured dynamically using the VARY TCPIP,,CMD=OBEYFILE command.

- Routing Performance Improvement

  To improve performance in reducing outbound RIP traffic, an option is provided in OE RouteD to broadcast the virtual network route only.

  Also, options are provided in OE RouteD to customize the interface scan and poll values to provide performance tuning in route switching for network recovery. The options are used to override the default time intervals to check for new interfaces, a new HOME list, and the status of interfaces.

Notes:

1. Point-to-point networks, such as CTC or IP over CDLC, cannot be configured as a primary or secondary interface. However, VIPA traffic can be routed over point-to-point networks.
2. VIPA allows other hosts to maintain connections to MVS when a network attachment fails. Assignment of choices of paths **into** MVS is accomplished by routing protocols out in the network. The assignment of primary and secondary physical links for direct MVS attachment to LANs allows OE RouteD to select alternate paths for **outbound** traffic when failures occur.

   As a general rule, the first link in each network or subnetwork is assigned as primary and others as secondary or backup.

RIP Input/Output Filters

The RIP input/output filters provide routing table manipulation and routing control. The filters are provided by OE RouteD and consist of:

1. Route Blocking (or NoReceiving)
2. Route Forwarding (Unconditional and Conditional)
3. Route Receiving (Unconditional and Conditional)
4. Route NoForwarding
5. Interface Broadcasting Switch
6. Interface RIP Switch
7. Default Route Only Broadcasting Switch
8. Virtual Route Only Broadcasting Switch
9. Default and Virtual Routes Only Broadcasting Switch
10. Triggered Updates Only Broadcasting Switch

For more information on these RIP input/output filters, see the OE RouteD procedure parameters in “OE Routed Parameters” on page 234 and the options statement in “Step 4: Configure the Gateways File or Data Set (Optional)” on page 230.

Using Virtual IP Addressing to Split Traffic

The purpose of splitting traffic is to reduce traffic load on network attachments. Since TCPIP for MVS and OE RouteD do not support load balancing, the traffic splitting techniques can be used to control inbound and outbound traffic. The following techniques can be used to produce traffic splitting effects with fault tolerance benefit:

- Using Interface Metric and VIPA To Split Inbound/Outbound Traffic

  In the primary/alternate network attachments to the same LAN configuration, split inbound/outbound traffic can be achieved by configuring the metric on the primary interface to one higher than the secondary interface(s). From routing updates, an adjacent router uses the gateway of a secondary interface to reach the destination VIPA on the OS/390 server because the route to the gateway has a shorter metric. The primary interface is used for outbound traffic and a secondary interface is used for inbound traffic. The traffic splitting will function as long as the primary and at least one secondary interfaces are active. For information on configuring an interface metric, see the “BSDROUTINGPARMS

Statement” on page 48. A VARY TCPIP,,CMD=OBEYFILE command for the BSDROUTINGPARMS statement can be used to update an interface metric for a link. For an example of configuring a virtual device, see “Configuring a Virtual IP Address” on page 238.

- Using Route Forwarding and VIPA to Split Session Traffic

  With multiple VIPAs in one TCP/IP stack, a VIPA can be assigned to a particular interface so that the VIPA can be reserved for session traffic (for example, FTP or TELNET). This is accomplished by using the route forwarding option in OE RouteD. From routing updates, an adjacent router will have multiple gateways to reach the VIPAs on the OS/390 server. The adjacent router will use one gateway to reach one VIPA reserved for one type of session traffic and the other gateway to reach another VIPA reserved for another type of session traffic on the OS/390 server. For fault tolerance, it is recommended that the conditional option of route forwarding be used. For information on route forwarding, see the options statement in in “Step 4: Configure the Gateways File or Data Set (Optional)” on page 230. For an example of configuring a virtual device, see “Configuring a Virtual IP Address” on page 238.

Using Virtual IP Addressing to Backup an OS/390 Server

Since a virtual IP address selects a TCP/IP stack (and MVS image if there is only one stack on an MVS image) and does not select a specific network attachment, it can be backed up to another TCP/IP stack on one MVS image or to another OS/390 server. This allows other hosts that were connected to MVS TCP/IP applications on the primary OS/390 server or TCP/IP stack to reestablish sessions with a backup OS/390 server or TCP/IP stack using the primary OS/390 server's or TCP/IP stack's virtual IP address. After the primary OS/390 server or TCP/IP stack has been restored, the temporarily reassigned virtual IP address can be restored.

Consider the following when backing up and restoring an OS/390 server or TCP/IP stack:

- All sessions between the clients and the server on the failing host will be disrupted.
- The client can use any ephemeral port number when re-establishing the connection to the backup server.
- Having a different port number for the backup and primary server is not recommended. If the backup server has a different port number than the original primary server (for example, port 101 rather than port 21 for FTP), the client must know to use 101 rather than 21. Different port numbers does work, but can cause administrative problems.

For more information on backing up and restoring an OS/390 server or TCP/IP stack, see “Configuring a Backup OS/390 Server with VIPA” on page 241 and “Restoring a Primary OS/390 Server with VIPA” on page 242.

OE RouteD Static Routes

In general, two types of static routes can be added to the OE RouteD gateways file or data set:

Passive Known by TCP/IP and OE RouteD.
External Known by OE RouteD, but not by TCP/IP.

Passive Routes

Information about passive routes is put in TCP/IP's and OE RouteD's routing tables. A passive entry in OE RouteD's routing table is used as a placeholder to prevent a route from being propagated and from being overwritten by a competing RIP route. With the exception of directly-connected passive routes, passive routes are not propagated; they are known only by this router. Using passive routes can create routing loops, so they need to be created carefully.

Defining passive routes such as these should be avoided:

A to C is via B.
B to C is via A.

Passive routes should be used when adding routes where the host/net is not running RIP. Passive routes should also be used when adding a default route, since this is the only way to prevent a route from timing out.

External Routes

External routes, such as the External Gateway Protocol (EGP), are managed by other protocols. The OE RouteD server needs to know not to interfere with these and not to delete them.

An external entry exists in OE RouteD's routing tables as a place holder to prevent a route from being overwritten by a competing RIP route. External routes are not propagated. OE RouteD does not manage external routes. Therefore, OE RouteD only knows that there is an existing route to host/net and one that is known to TCP/IP.

External routes should be used when the local host is running with some type of non-RIP routing protocol which dynamically changes the TCP/IP routing tables. The foreign host does not need to run any routing protocol, since the only concern is how to route traffic from the local host to the foreign host, and how to prevent multiple routing protocols from interfering with each other.

OE RouteD Active Gateways

In general, active routes can be added to the OE RouteD gateways file or data set. An active route is treated as a network interface.

Active gateways are routers which are running RIP, but are reached by a medium which does not allow broadcasting and is not point-to-point. OE RouteD normally requires that routers be reachable via broadcast addresses or via a point-to-point link. If the interface is neither, then an active gateway entry can add the gateway to OE RouteD's interface list. OE RouteD will treat the active gateway as a network interface. Note that the active gateway must be directly connected.

Active routes should be used when the foreign router is reachable over a non-broadcast and non-point-to-point network, and is directly connected to the local host.

OE RouteD will communicate with active routers by point-to-point transmissions to the gateway address. Routes are not added to either OE RouteD or the TCP/IP routing table immediately. They are added and propagated normally when route advertisements arrive from an active gateway. The sole effect of an active gateway statement is to bypass the requirement for broadcast communication on true point-to-point links. Interfaces which are not broadcast, not point-to-point, and are not active gateways are assumed to be loopback interfaces to the local host. Also, while a route to an active gateway might time out, the interface entry is never removed. If transmissions resume, then the new routes will still be available to the active gateways.

OE RouteD Gateway Summary

Table 11. OE RouteD Gateway Summary

	Propagate	TCP/IP	RouteD	Timeout
Dynamic [1]	Yes	Yes	Yes	Yes
Passive	No [2]	Yes	Yes	No
External	No	No	Yes	No
Active	Yes	Yes	Yes	Yes

Configuration Process

The steps to configure OE RouteD are as follows:

1. Update PORT, BSDROUTINGPARMS, GATEWAY, and IPCONFIG statements in the TCPIP profile
2. Update the resolver configuration file
3. Update the OE RouteD cataloged procedure (optional)
4. Specify the OE RouteD port number in the SERVICES file or data set
5. Configure the gateways file or data set (optional)

   Note: If a default route is to be defined to a destination gateway or router, configure a default route in this gateways file or data set.

[1] Dynamic routing is provided by OE RouteD.

[2] Except directly-connected passive routes. Directly-connected passive routes are propagated to other network interfaces for network reachability. A directly-connected passive route is one where the gateway address is one of the local interfaces in the HOME list or is one of the offload interfaces.

Step 1: Specify Configuration Statements in PROFILE.TCPIP

To ensure that UDP port 520 is reserved for OE RouteD, also add the name of the member containing the OE RouteD cataloged procedure to the PORT statement in PROFILE.TCPIP:

```
PORT
   520 UDP OROUTED
```

In addition, configure the BSDROUTINGPARMS statements with your routing information. The GATEWAY statement is not used and should be removed or commented from PROFILE.TCPIP.

Code the following on the IPCONFIG statement in PROFILE.TCPIP:

```
IPCONFIG IGNOREREDIRECT DATAGRAMFWD
```

Do not specify the no forwarding (NOFWD) option.

See Chapter 3, “Configuring the TCPIP Address Space” on page 33 for descriptions and examples of these statements.

Note: If you want to be able to start orouted from the OE shell, use the special name OMVS as follows:

```
PORT 520 UDP OMVS
```

This enables the entire “OMVS job group” (that is, all OE shell users). The shell users must be superuser for this to work.

Step 2: Update the resolver configuration file

The resolver configuration file or data set contains keywords that are used by OE RouteD. Two important keywords in the resolver file are DATASETPREFIX and TCPIPjobname. The value assigned to DATASETPREFIX will determine the high-level qualifier (*hlq*). The *hlq* is then used in the search order for other configuration files. If no DATASETPREFIX keyword is found in the resolver configuration dataset or file, a default of TCPIP is used. In a CINET environment, the value assigned to TCPIPjobname will be used as the name of the stack with which OE RouteD attempts to establish a connection. In an INET environment, it is not necessary to set TCPIPjobname, but if it is set, it must be set to "INET". See Figure 5 on page 228 for a sample resolver configuration file. The resolver uses the following search order to locate the actual resolver configuration data set or file to use:

1. If the environment variable RESOLVER_CONFIG has been defined, the resolver uses the value of this environment variable as the name of an MVS data set or HFS file to access the resolver configuration data. The syntax for an MVS data set name is *"//'mvs.dataset.name'"*. The syntax for an HFS file name is *"/dir/subdir/file.name"*.
2. /etc/resolv.conf
3. Any MVS data set that is pre-allocated to a DD-name of SYSTCPD. Due to restrictions for DD-name allocations during fork() processing, the use of this technique is not encouraged.
4. *userid*.TCPIP.DATA for TSO/E or *jobname*.TCPIP.DATA for a batch request
5. SYS1.TCPPARMS(TCPDATA)

```
;*********************************************************************
;   Name of File:           TCPIP.DATA                                *
;   This data set, TCPIP.DATA, is used to specify configuration       *
;   information required by TCP/IP client programs.                   *
;   Syntax Rules for the TCPIP.DATA configuration data set:           *
;   (a) All characters to the right of and including a ';' will be    *
;       treated as a comment.                                         *
;   (b) Blanks and <end-of-line> are used to delimit tokens.          *
;   (c) The format for each configuration statement is:               *
;       <SystemName||':'>  keyword  value                             *
;       where <SystemName||':'> is an optional label which may be     *
;       specified before a keyword; if present, then the keyword-     *
;       value pair will only be recognized if the SystemName matches  *
;       the node name of the system, as defined in the IEFSSNxx       *
;       PARMLIB member.  This optional label permits configuration    *
;       information for multiple systems to be specified in a single  *
;       TCPIP.DATA data set.                                          *
;*********************************************************************
; TCPIPuserid specifies the userid of the TCPIP address space.
; TCPIP is the default userid.
TCPIPjobname TCPV33A
; HostName specifies the TCP host name of this system.  If not
; specified, the default HostName will be the node name specified
; in the IEFSSNxx PARMLIB member.
MVS5: HostName MVS5
DATASETPREFIX tcpv33a
; DomainOrigin specifies the domain origin that will be appended
; to host names passed to the resolver.  If a host name contains
; any dots, then the DomainOrigin will not be appended to the
; host name.
DomainOrigin TCP.RALEIGH.IBM.COM
; NSinterAddr specifies the internet address of the Name Server.
; Multiple Name Server addresses may be specified.  The Name Servers
; will be tried in the given order.
NSinterAddr  9.37.32.94
; NSportAddr specifies the Name Server port.
; 53 is the default value.
NSportAddr 53
; ResolveVia specifies how the Resolver is to communicate
; with the Name Server. TCP indicates use of TCP virtual circuits.
; UDP indicates use of UDP Datagrams.
; The default is UDP.
ResolveVia UDP
; ResolverTimeout specifies the time in seconds that the Resolver
; will wait while trying to open a TCP connection to the name server,
; or how long it will wait for a response when using UDP.
ResolverTimeout 30
; ResolverUdpRetries specifies the number of times the resolver should
; retry a query to the Nameserver when using UDP datagrams.
ResolverUdpRetries 1
MVS1: MESSAGECASE UPPER
; End of file.
```

Figure 5. Sample resolver configuration file

Step 3: Update the OROUTED Cataloged Procedure (optional)

If OE RouteD is to be started by a procedure, update the cataloged procedure OROUTED by copying the sample in *hlq*.SEZAINST(OROUTED) to your system or recognized PROCLIB. Specify OE RouteD parameters and change the data set names as required to suit your local configuration.

OE RouteD Cataloged Procedure (OROUTED)

```
//*****************************************************************
//OROUTED  EXEC PGM=BPXBATCH,REGION=4096K,TIME=NOLIMIT,
//      PARM='PGM /usr/sbin/orouted -t -t -ep'
//*
//* The STDOUT statement specifies the file that will contain
//* the standard output of OE RouteD.  The -ep parameter must be specified so
//* the standard output will be enabled.
//STDOUT   DD PATH='/tmp/routed.stdout',
//         PATHOPTS=(OWRONLY,OCREAT,OAPPEND),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//*
//STDERR   DD PATH='/tmp/routed.stderr',
//         PATHOPTS=(OWRONLY,OCREAT,OAPPEND),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//*
//* The STDENV statement allows environment variables to be defined for
//* use by OE RouteD.  Examples of the contents of this file are as follows:
//*       RESOLVER_CONFIG=/etc/resolv.conf
//*       GATEWAYS_FILE=/etc/gateways.tcpv33a
//*
//STDENV   DD PATH='/u/user105/release/orouted.env'
//*
//SYSERR   DD PATH='/tmp/routed.log',PATHOPTS=(OWRONLY,OCREAT,OAPPEND),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//*
```

Step 4: Update SERVICES File

The services data set or file contains the relationship between service names (servers) and port numbers in the OpenEdition MVS environment. The portion of the services file relevant to OE RouteD is shown in Figure 6. The data set or file must exist for OE RouteD to run. The following search order is used to find the services data set or file:

1. /etc/services
2. *userid*.ETC.SERVICES for TSO/E or *jobname*.ETC.SERVICES for a batch request
3. *hlq*.ETC.SERVICES

```
# Start of IBM added services ...
route        520/udp       router routed
```

Figure 6. Sample Portion of Services File

Step 4: Configure the Gateways File or Data Set (Optional)

The OE RouteD server queries the network and dynamically builds routing tables from routing information transmitted by other routers that are directly connected to the network. The gateways file or data set is used to further configure the routing tables.

Note: The gateways file or data set is not related to the GATEWAY statement used in the PROFILE.TCPIP data set.

The OE RouteD server uses the following search order to locate the GATEWAYS configuration data set or file:

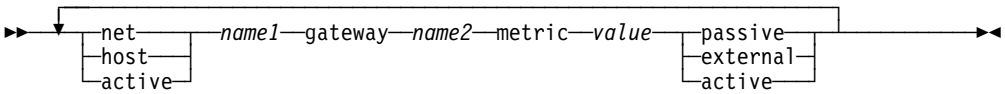
1. If the environment variable GATEWAYS_FILE has been defined, OE RouteD uses this value as the name of an MVS data set (//'mvs.dataset.name') or HFS file (/dir/subdir/file.name) to access the gateways file
2. /etc/gateways
3. *hlq*.ETC.GATEWAYS

A `passive` entry in the gateways file or data set is used to add a route to a part of the network that does not support RIP. An `external` entry in the gateways file or data set indicates a route that should never be added to the routing tables. If another RIP server offers this route to your host, the route is discarded and not added to the routing tables. An `active` entry indicates a gateway that can only be reached through a network that does not allow or support broadcasting.

Syntax Rules

- Keywords can be specified in mixed case.
- Blanks and comments are supported in the gateways file or data set. Comments are identified by a semicolon in column 1.
- There should be no sequence numbers in the data set.

The syntax for the gateways file or data set is:



net
Indicates the route goes to a network.

host
Indicates the route goes to a specific host.

active
Indicates that the route to the gateway will be treated as a network interface.

name1
Can be either a symbolic name or the IP address of the destination network or host. If an IP address is specified, it must be in the standard dotted decimal notation. All numbers will be interpreted as decimal values only. No hexadecimal nor octal notation will be accepted.

name1 must be specified as “active” if this is for an active gateway. The last entry in the data set must specify an active gateway.

gateway
A constant. The parameters that follow this keyword identify the gateway or router for this destination.

name2
Can be either a symbolic name or the IP address of the gateway or router for this destination. If an IP address is specified, it must be in the standard dotted decimal notation. All numbers will be interpreted as decimal values only. No hexadecimal nor octal notation will be accepted.

metric
A constant. The value that follows this keyword is the hop count to the destination host or network.

value
The hop count to this destination. This number is an integer in the range of 0 through 16, where 16 (infinity) indicates the network cannot be reached.

passive
A passive gateway does not exchange routing information. Information about the passive gateway is maintained in the local routing tables indefinitely and is only local to this OE RouteD server. Passive gateway entries are not included in any routing information that is transmitted. Directly connected passive routes are included.

external
An external gateway parameter indicates that entries for this destination should never be added to the routing table. The OE RouteD server discards any routes for this destination that it receives from other routers. Only the destination field is significant. The gateway and metric fields are ignored.

active
Active gateways are treated as network interfaces. Active gateways are routers that are running RIP, but can only be reached through a network that does not allow broadcasting and is not point-to-point.

Note: For more information on passive, external, and active gateways, see "OE RouteD Static Routes" on page 225.

The following example shows the contents of a gateways file or data set containing multiple entries:

```
net    acmenet       gateway  gateway.acme.com  metric 5  passive
host   vm3.ibm.com   gateway  9.67.43.126       metric 6  passive
host   bad.host      gateway  xxx               metric 1  external
active active        gateway  9.3.1.110         metric 3  active
net    0.0.0.0       gateway  9.67.112.1        metric 1  passive
```

In the first entry, the route indicates that `acmenet` can be reached through the gateway `gateway.acme.com`, and that it is 5 hops away.

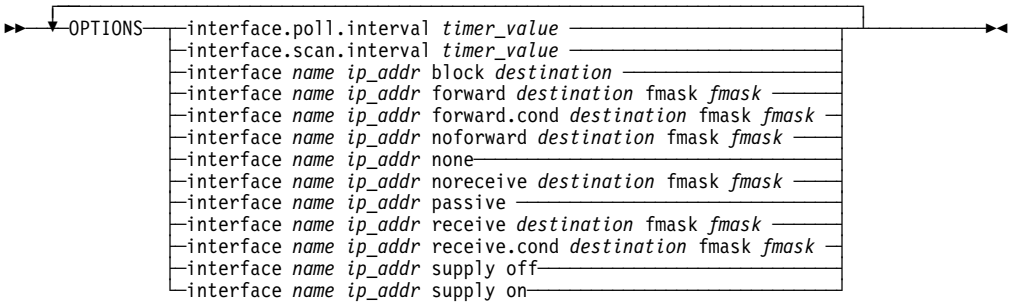
In the second entry, the route indicates that `vm3.ibm.com` can be reached through the gateway `9.67.43.126`, and that it is 6 hops away.

In the third entry, the external gateway parameter indicates that routes for the host `bad.host` should not be added to the routing tables, and that routes received from other OE RouteD servers for `bad.host` should not be accepted.

The fourth entry shows an active gateway.

The fifth entry shows a default route to the destination gateway 9.67.112.1.

The syntax for the OPTIONS statement for the gateways file or data set is:



interface.scan.interval
Specifies the time interval in seconds for the interface scan interval. OE RouteD uses this timer value to rescan existing interfaces for up/down status, new interfaces, and new HOME lists. New interfaces and HOME lists are dynamically added using VARY TCPIP,,CMD=OBEYFILE commands.

timer_value The range is from 30 to 180 seconds in multiples of 30 seconds. The default is 60 seconds.

interface.poll.interval
Specifies the time interval in seconds for the interface poll interval. OE RouteD uses this timer value to check existing interfaces for up/down status only. Triggered updates are issued during interface outages to inform adjacent routers of unreachable routes so that alternative routes can be discovered.

timer_value The range is from 15 to 180 seconds in multiples of 15 seconds. The default is 30 seconds.

interface
A constant

name
Specifies the name of the interface as used in the HOME list.

ip_addr
Specifies the internet address of the interface associated with the interface name.

block
Specifies that the *destination* route in the received broadcasts for this interface is to be ignored. This option is provided as a RIP input filter.

destination
Specifies the destination route in network, subnetwork, or host format.

fmask
Specifies the optional route filter mask.

forward
Specifies that the *destination* route in the routing table broadcasts is to be forwarded to this interface only. This option is provided as a RIP output filter and can be used for inbound and outbound traffic splitting.

forward.cond
Specifies that the *destination* route in the routing table broadcasts is to be forwarded to this interface only when the interface is active. In case of an interface outage, OE RouteD will include the *destination* route in the routing table broadcasts to other active interfaces. This option is provided as a RIP output filter and can be used for inbound and outbound traffic splitting.

noforward
Specifies that the destination route in the routing table is not to be forwarded. This option is provided as a RIP output filter.

noreceive
See description for block.

passive
Specifies that RIP is disabled for this interface. OE RouteD will not broadcast and will ignore routing updates. This option is provided as a RIP input and output filter.

receive
Specifies that the *destination* route is to be received over this interface only. If it is received over any other interface, the route is discarded. This option is provided as a RIP input filter.

receive.cond
Specifies that the *destination* route is to be received over this interface only when the interface is active. In case of an interface outage, OE RouteD will include the *destination* route in the routing table broadcasts to other active interfaces. This option is provided as a RIP input filter.

supply off
Specifies that broadcasting is disabled for this interface. OE RouteD will not broadcast, but continues to receive routing updates. This option is provided as a RIP output filter.

supply on
Specifies that broadcasting is enabled for this interface. This option is provided as a RIP output filter.

none
Specifies that any RIP filter options for this interface are to be turned off or reset.

The following example shows the options entries of a gateways file or data set:

```
options interface.scan.interval 90
options interface.poll.interval 15
options interface ETH1 10.1.1.1 passive
options interface ETH1 10.1.1.1 supply off
options interface TR1 9.67.112.25 forward 11.0.0.0
options interface TR1 9.67.112.25 forward.cond 12.0.0.0
options interface TR1 9.67.112.25 block 9.1.0.0
```

OE Routed Parameters

OE Routed accepts the command line parameters listed below. These parameters are valid when starting the program from either an MVS procedure or from the OE shell. They are also valid when modifying OE RouteD with the MODIFY command. For information on using the MODIFY command, see"MODIFY Command—OE RouteD Server" on page 242

-d Enables printing internal debug information to STDOUT. This option should only be used to debug problems. When this option is specified, the -ep parameter is set internally.

-dp Traces packets to and from adjacent routers and received and broadcasted RIP network routing tables. Packets are displayed in data format. Output is written to STDOUT.

-g Enables the default router. When this option is specified, OE RouteD will add a default route to its routing information and broadcast it over all local interfaces. If the adjacent routers add the default route to their routing tables, OE RouteD will receive all unknown packets from them and funnel them to a destination router, provided that a default route is defined. If you use this option, we recommend that you define a default route to a destination router in the gateways file or data set. See "Configuring a Default Route" on page 238.

Note: Do not use this option if default routes are to be learned dynamically from adjacent routers.

-h Include host routes in the routing information for the broadcasts. Adjacent routers should support Host Route Broadcasting to prevent NETWORK UNREACHABLE problems from occurring.

-hv Include only virtual host routes in the routing information for the broadcasts. Adjacent routers should support Host Route Broadcasting, or network or subnetwork portions of VIPA addresses must be unique for each TCP/IP image.

-q Suppresses broadcasting of routing information.

-sd Supply default route only. When this option is specified, the -g parameter is set internally. This option is provided as a RIP output filter.

-sdv (or -svd)
Supply virtual routes and default routes only. See parameter descriptions for -sv and -sd. This option is provided as a RIP output filter.

-st Supply triggered updates only. Similar to the -q parameter except that OE RouteD will supply network unreachable routing information during interface outages so that adjacent routers can recover by switching to different routes rather than relying on three-minute timeouts. This option is provided as a RIP output filter.

-sv Supply virtual routes only. Recommended usage is when multiple network adapters in a TCPIP stack are in the same network; otherwise, network connectivity problems will occur. This option is provided as a RIP output filter.

-svd Similar to -sdv parameter.

-ep Enable display of program print statements to STDOUT and STDERR. Information can be saved to a file by redirecting STDOUT to a file using the '>' operator. If this option is specified, and the program is started in the background from an OE shell, the userid will not be able to exit the shell until the program has ended.

-t Activates tracing of actions by the OE RouteD server.

-t -t Activates tracing of actions and packets sent or received.

-t -t -t Activates tracing of actions, packets sent or received, and packet history. Circular trace buffers are used for each interface to record the history of all packets traced and are displayed whenever an interface becomes inactive.

-t -t -t -t Activates tracing of actions, packets sent or received, packet history, and packet contents. The packet displays the RIP network routing information.

Specifying Parameters

If OE RouteD is to be started from an MVS procedure, add your parameters to `PARM='PGM` in the OROUTED cataloged procedure, making certain that a slash precedes the first parameter. For example: `//PARM='PGM /usr/sbin/orouted /-g -q -t -t -ep'`

If OE RouteD is to be started from an OE shell command line, enter the parameters on the OE shell command line.

For either method of starting OE RouteD, the following apply:

- Each parameter is separated by a blank
- Parameters can be specified in mixed case.

Starting OE RouteD

In a CINET environment, OE RouteD will attempt to connect to a stack name whose name is determined by the *TCPIPJobname* keyword from the resolver configuration data set or file. The *TCPIPJobname* must match the NAME field for ENTRYPOINT(EZBPFINI) in the BPXPRMxx member you used to start OMVS. For information on configuring mulitple TCP/IP instances as OpenEdition CINET physical file systems, see “Considerations for Multiple Instances of TCP/IP” on page 26.

In configurations with multiple stacks, a copy of OE RouteD must be started for each stack that requires OE RouteD services. To associate OE RouteD with a particular stack, use the environment variable RESOLVER_CONFIG to point to the data set or file that defines the unique *TCPIPjobname*. A unique gateways file can be associated with each copy of OE RouteD by defining the environment variable GATEWAYS_FILE. In the example in Figure 7 on page 236 for the OE shell, there are two active stacks with the names TCPV33A and TCPV33B. First the environment variable RESOLVER_CONFIG is set to point to the file that defines the *TCPIPjobname* TCPV33A, then the environment variable GATEWAYS_FILE is set to point to the gateways file /etc/gateways.tcpv33a, and then OE RouteD is started for that stack. Next, RESOLVER_CONFIG is set to point to another configuration

file that defines *TCPIPjobname* TCPV33B, then the environment variable GATEWAYS_FILE is set to point to the gateways file /etc/gateways.tcpv33b, and then ORouteD is started for that stack.

```
#  export RESOLVER_CONFIG=/u/user105/tcpv33a.conf            !point to TCPV33A resolver file
#  export GATEWAYS_FILE=/etc/gateways.tcpv33a                !point to TCPV33A gateways file
#  orouted&                                                  !start orouted in the background
#  export RESOLVER_CONFIG=/u/user105/tcpv33b.conf            !point to TCPV33B resolver file
#  export GATEWAYS_FILE=/etc/gateways.tcpv33b                !point to TCPV33B gateways file
#  orouted&                                                  !start orouted in the background
```

Figure 7. Example Commands to Start Multiple Copies of ORouteD

When running from an MVS procedure, the environment variables can be set by using the STDENV DD statement in the procedure used to start OE RouteD. For an example of using the STDENV DD statement, see "OE RouteD Cataloged Procedure (OROUTED)" on page 229.

Configuration Examples

This section contains examples for configuring the OE RouteD server. The following example illustrates a OE RouteD configuration.

Configuring a Passive Route

In Figure 8, assume that your OS/390 server is `host1` and is running an OE RouteD server. The other two hosts, `host2` and `host3`, are not running a RIP server. Your OE RouteD server does not learn a route to `host3`, because `host2` is not running a RIP server. Your OE RouteD server sends routing updates to `host3` over the link to `host2` but never receives a routing update from `host2`. After 180 seconds, your OE RouteD server deletes the route to `host2`. This problem is inherent to the RTP protocol and cannot be prevented.

To solve the problem, you should add a `passive` route to this host in the gateways file or data set. You can use either of the following gateway statements:

```
host  host3        gateway  host2        metric 2  passive

host  192.10.10.2  gateway  192.10.20.2  metric 2  passive
```

Similarly, if `host2` is not running a RIP server, you can define a directly-connected `passive` route as follows:

```
host  host2        gateway  host1        metric 1  passive
```

A directly-connected passive route is one where the gateway address or name is one of the local interfaces in the HOME list or is one of the offload interfaces.

Assume that your OS/390 server is now `host2` and is running a OE RouteD server. `host1` is also running a RIP server, but `host3` is not. Your OE RouteD server sends routing information updates to `host3` over the link to `host3` but never receives a routing update from `host3`. After 180 seconds, your OE RouteD server deletes the route to `host3`.

You should add a passive route to this host as follows:

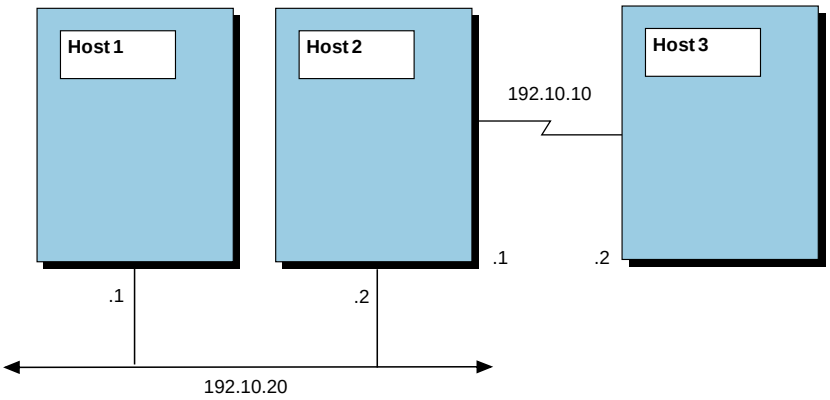


Figure 8. RouteD Configuration Example

```
host  host3        gateway  host2        metric 1  passive
```

host1 cannot reach host3 unless a passive routing entry is added to host1. For example:

```
host   host3    gateway  host2  metric  2  passive
```

or

```
host  192.10.10.2  gateway  192.10.20.2  metric  2  passive
```

Configuring an External Route

In Figure 8, assume that your OS/390 server is again host1, which is running an OE RouteD server. The other two hosts, host2 and host3, are also running RIP servers. Your OE RouteD server normally learns a route to host3 from host2, because host2 is running a RIP server. You might not want host1 to route to host3 for security reasons. For example, a university might want to prevent student hosts from routing to administrative hosts.

To prevent your OE RouteD server from adding a route to host3, add an external route to the gateways file or data set. You can use either of the following gateway statements:

```
host  host3        gateway  host2        metric 2  external

host  192.10.10.2  gateway  192.10.20.2  metric 2  external
```

Configuring a Point-to-Point Link

The OE RouteD server can manage point-to-point links that have a routed server on the other end of the link. For example, if 2 VM or MVS TCP/IP hosts are connected by a point-to-point link, then OE RouteD can be used to manage the link ***only*** if both hosts are running a routed server. If only one of the hosts is running a routed server, then passive routing (GATEWAY statements) must be used to configure the routing for the link. See “Configuring a Passive Route” on page 236.

Configuring a Default Route

A default route is typically used on a gateway or router to an internet, or on a gateway or router that uses another routing protocol, whose routes are not reported to other local gateways or routers.

To configure a route to a default destination, add a default route using the passive route definition in the gateways file or data set. For example, if the default destination router has a gateway address 9.67.112.1, then add the following entry to the data set:

```
net  0.0.0.0  gateway  9.67.112.1  metric  1  passive
```

Only one default route to a destination gateway or router can be specified. OE RouteD currently does not support multiple default routes.

Configuring a Virtual IP Address

VIPA provides an IP address that selects a TCP/IP image (and MVS system if there is only one image on an MVS system) without selecting a specific network attachment. Other hosts that connect to MVS TCP/IP applications can send data to an MVS VIPA using those paths selected by the routing protocols. VIPA provides tolerance of failures of MVS network attachment hardware.

Assume that you want to configure two virtual IP addresses in one TCP/IP image in Figure 9 on page 239.

Include the following DEVICE and LINK statements in PROFILE.TCPIP:

```
DEVICE VDEV1  VIRTUAL 0
LINK   VLINK1 VIRTUAL 0 VDEV1
DEVICE VDEV2  VIRTUAL 1
LINK   VLINK2 VIRTUAL 0 VDEV2
```

Add the virtual link to the HOME statement in PROFILE.TCPIP:

```
HOME
  9.2.1.1   VLINK1
  9.3.1.1   VLINK2
```

Update the BSDROUTINGPARMS statement in PROFILE.TCPIP:

```
BSDROUTINGPARMS false
   VLINK1    DEFAULTSIZE   0   255.255.0.0   0
   VLINK2    DEFAULTSIZE   0   255.255.0.0   0
ENDBSDROUTINGPARMS
```

Code the following on the IPCONFIG statement in PROFILE.TCPIP:

```
IPCONFIG IGNOREREDIRECT DATAGRAMFWD
```

Notes:

1. Any real links defined in the OS/390 server must be defined in addition to the virtual links.
2. Do not specify the NOFWD option.

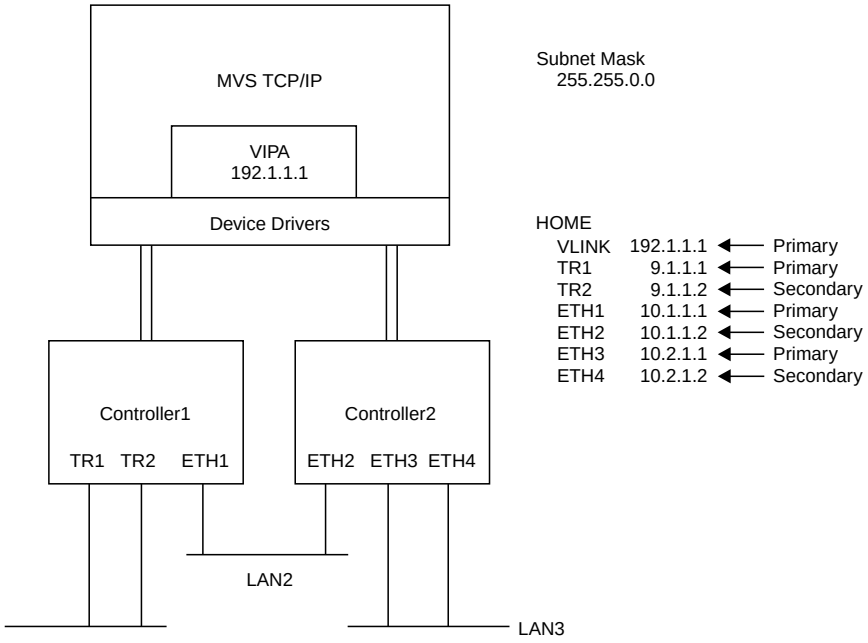


Figure 9. Multiple Network Attachments Configuration. Sample configuration showing primary/multiple network attachments to LAN segments on one controller and across two controllers.

Note the primary/secondary assignments according to the order of the HOME list. For virtual links, primary assignment is made to the first virtual IP address in the HOME list or to the one from the PRIMARYINTERFACE statement. The remaining virtual IP addresses are assigned as secondary.

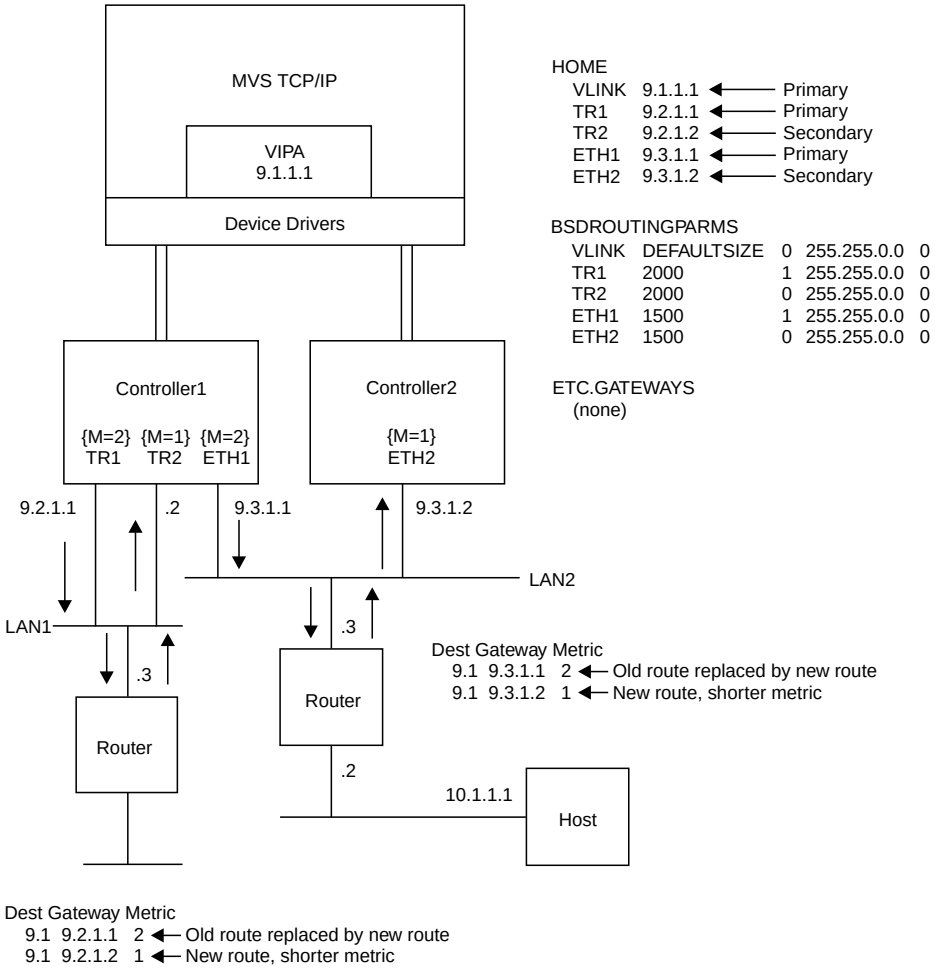


Figure 10. Single VIPA Configuration. Sample configuration showing primary/multiple network attachments to the same LAN, VIPAs, and inbound/outbound traffic splitting.

Note that the cost-of-use metrics for TR1 and ETH1 primary interfaces are changed. The metric values in the BSDROUTINGPARMS statement are incremented by one and are added to the routing metrics in the outbound RIP packets. As a result, outbound traffic occurs on interfaces TR1 and ETH1 and inbound traffic occurs on interfaces TR2 and ETH2. Traffic splitting can occur on one controller (for example, interfaces TR1 and TR2) as well as across multiple controllers (for example, interfaces ETH1 and ETH2).

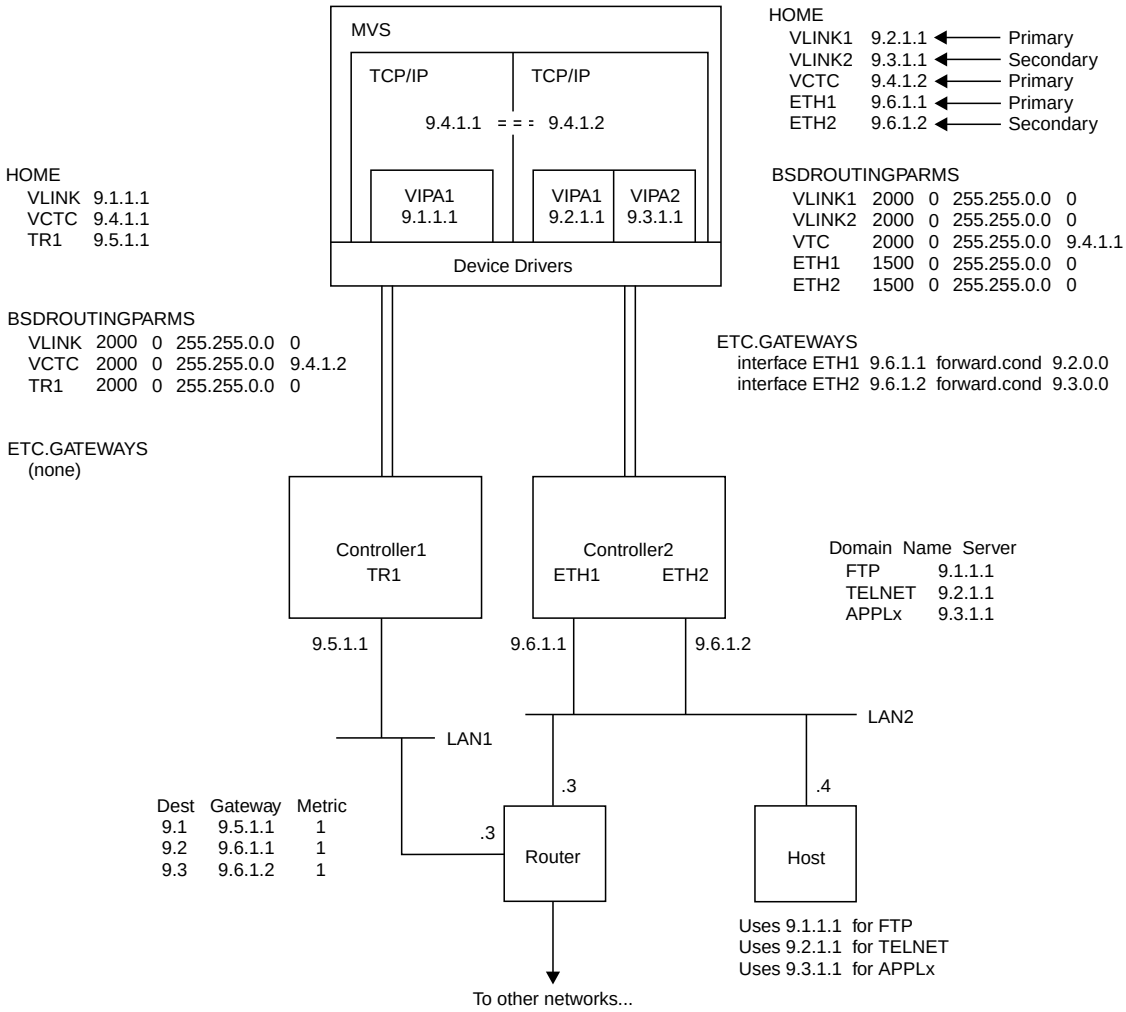


Figure 11. Multiple VIPAs Configuration. Sample configuration showing primary/multiple network attachments to the same LAN, VIPAs, and session traffic splitting.

Note that route forwarding with conditional option is used for interfaces ETH1 and ETH2. Outbound traffic flows on interfaces TR1 and ETH1; inbound traffic flows on interfaces TR1, ETH1, and ETH2 to the VIPA addresses. Inbound traffic splitting for the TCP sessions occurs on interfaces ETH1 and ETH2. The VCTC link between the two TCP/IP images is a point-to-point connection and is used to ensure connectivity to the VIPA addresses in case of controller failures. Also, note that route forwarding is not needed for interface TR1 because there is only one real adapter associated with controller 1 for VIPA1 (9.1.1.1).

Configuring a Backup OS/390 Server with VIPA

To configure a backup OS/390 server with the primary OS/390 server's or TCP/IP image's virtual IP address, do the following after a primary OS/390 server or TCP/IP image is down:

- If OE RouteD is running on the backup OS/390 server or TCP/IP image, issue VARY TCPIP,,CMD=OBEYFILE commands to:

  1. Add new virtual link and device statements.

2. Add new HOME and BSDROUTINGPARMS statements for the new virtual link. The HOME statement will have the primary OS/390 server's or TCP/IP image's virtual IP address. Ensure that the HOME and BSDROUTINGPARMS statements are defined in one VARY TCPIP,,CMD=OBEYFILE command.

- If OE RouteD is not running on the backup OS/390 server or TCP/IP image, restart OE RouteD and add the new virtual LINK and DEVICE, HOME, and BSDROUTINGPARMS statements using the VARY TCPIP,,CMD=OBEYFILE command.

Restoring a Primary OS/390 Server with VIPA

To restore a virtual IP address to the primary OS/390 server after it is no longer needed on the backup, do the following:

- If OE RouteD is running on the backup OS/390 server or TCP/IP image, issue a VARY TCPIP,,CMD=OBEYFILE command to add a new HOME statement with the primary OS/390 server or TCP/IP image's virtual IP address removed. Otherwise, restart OE RouteD, and using a VARY TCPIP,,CMD=OBEYFILE command, add a new HOME statement with the primary virtual IP address removed. It is not necessary to add a new BSDROUTINGPARMS statement.
- Start OE RouteD on the primary OS/390 server or TCP/IP stack after ensuring that the virtual LINK and DEVICE, HOME, and BSDROUTINGPARMS statements are defined for the virtual IP address that was temporarily reassigned to the backup OS/390 server or TCP/IP image.

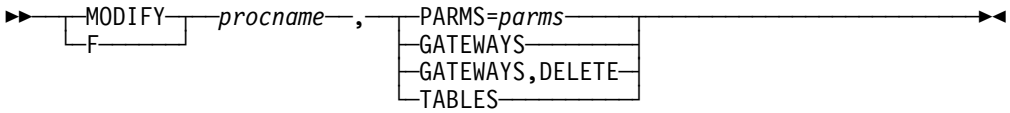
Controlling OE RouteD with the MODIFY Command

You can control most of the OE RouteD server functions from the operator's console using the MODIFY command. The following is the syntax and valid parameters.

MODIFY Command—OE RouteD Server

Use the MODIFY command to pass parameters to the OE RouteD server.

Syntax



Parameters

procname

If OE RouteD was started from a cataloged procedure, procname is the member name of that procedure. If OE RouteD was started from the OE shell, the procname is *userid*X, where *X* is the sequence number set by the system. To determine the sequence number, from the SDSF LOG window on TSO issue "/d omvs,u=*userid*". This will show the programs running under the userid *userid*.

parms

Any one or more of the following separated by a space.

-d Enables printing of internal debug information to STDOUT.

-dp Trace packets to and from adjacent routers and received and broadcasted RIP network routing tables. Packets are displayed in data format. Output is to STDOUT.

-dq Disable all debug traces.

-f Flush all indirect routes known by OE RouteD from IP routing tables.

-fh Flush all indirect host routes known by OE RouteD from IP routing tables.

-g Enable default route broadcasting. When this option is specified, OE RouteD will add a default route to its routing information and broadcast it over all local interfaces.

-gq Disable default route broadcasting.

-h Include host routes in the routing information for the broadcasts.

-hq Disable host route and virtual host route broadcasting.

-hv Include only virtual host routes in the routing information for the broadcasts

-hvq Disable virtual host route broadcasting.

-k Kill OE RouteD. OE RouteD will post a message to the console and to STDERR and then end.

-s When orouted is started with *-q,* use *-s* to force supply of routing information.

-sd Enable supply default route.

-sdv Similar to svd parameter.

-sdvq Similar to svdq parameter.

-sq or -q Disable supply all routes.

-st Supply triggered updates only

-stq Disable supply triggered updates

-sv Enable supply virtual route.

-svq Disable supply virtual route.

-svd (or sdv) Enable supply virtual and default routes.

-svdq (or sdvq) Disable supply virtual and default routes.

-t Enable or disable traces. Up to 4 -t parms are allowed.

-tq Disable all traces.

GATEWAYS

Reread the GATEWAYS file or data set.

GATEWAYS,DELETE

Reread the GATEWAYS file or dataset and delete all routes listed.

TABLES
Display RIP routing and interface tables (internal to OE RouteD).

Examples

Both of the following commands would pass parameters to an OE RouteD server started with a procedure named OROUTED.

```
MODIFY OROUTED,PARMS=-t -t -s
F OROUTED,GATEWAYS,DELETE,PARMS=-sq
```

Chapter 11. Configuring the OE PORTMAP Address Space

Before You Configure...:

Read and understand Chapter 1, “Before You Begin” on page 3. It covers important information about data set naming and search sequences.

This chapter describes how to configure the OE PORTMAP address space, which runs the Portmapper function.

Configuration Process

Steps to configure OE PORTMAP:

1. Specify PORT statements in *hlq*.PROFILE.TCPIP
2. Update the PORTMAP cataloged procedure

Step 1: Specify the PORT statements in PROFILE.TCPIP

To ensure that port UDP 111 and TCP 111 is reserved for the OE PORTMAP server, add the name of the member containing the PORTMAP catalogued procedure to the PORT statement in *hlq*.PROFILE.TCPIP:

```
PORT
  111 UDP OMVS     ; OE Portmapper Server
  111 TCP OMVS     ; OE Portmapper Server
```

See “PORT Statement” on page 83 for more information about the PORT statement.

Step 2: Update the PORTMAP Cataloged Procedure

Update the PORTMAP cataloged procedure to suit your local conditions by copying the sample provided in *hlq*.SEZAINST(OPORTPRC) to your system or recognized PROCLIB and modifying it to suit your local conditions. Change the data set names as required.

PORTMAP Cataloged Procedure (OPORTPRC)

```
//PORTMAP PROC
//*
//*  OpenEdition MVS Portmapper Server main process
//*  Resulting address space name will be PORTMAP1, when
//*  we use this method to start the portmapper
//*
//*  OPORTMAP is in SYS1.TCPIP.SEZALINK (on the LINKLST)
//*
//PORTMAP EXEC PGM=OPORTMAP,REGION=40M,
//        PARM='POSIX(ON),ALL31(ON)/'
//STDOUT   DD SYSOUT=*
//STDERR   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//         PEND
```

Starting the PORTMAP Address Space

There are two ways to start the portmapper as an OpenEdition socket application:

- From the OE shell
- As a started task.

To start the portmapper from the OE shell, the user ID must be an authorized superuser. The authorized superuser ID can issue “oportmap &” to start the portmapper. For the authorization procedure, see OS/390 OpenEdition Planning.

You can also start PORTMAP as a started task with the START command as follows:

```
START PORTMAP
```

Note: If your system is using the Network File System (NFS) server, see *Customizing and Operating the Network File System Server* for more information.

Part 3. Appendixes

Appendix A. SMF Records

This appendix describes the SMF records for the OE FTP server.

FTP Server SMF Record Layout

The SMF record written by the FTP server has the following format:

Table 12. FTP Server SMF Record Format

Byte	Subfield (offset)	Description
0–23		Standard SMF header
	SMFxFLG (4)	A system indicator. If the first bit is ON, record subtypes are valid.
	SMFxRTY (5)	A record type that is set to 118 (X'76') for all TCP/IP records
	SMFxSTY (22)	The record subtype obtained from the SMF statements in the FTP.DATA data set. The subtype value is in the range X'0' – X'FF'.
24–27		4-byte FTP subcommand (for example, STOR, REN, DELE)
28–31		4-byte FTP file type (SEQ, JES, SQL)
32–35		Fullword client IP address
36–39		Fullword server IP address
40–47		Reserved
48–55		Local user ID
56		Data format (A—ASCII, E—EBCDIC, and so on)
57		Mode (S—stream , B—block, C—compressed)
58		Structure (F—file)
59		Data set type (P—partitioned, blank—sequential, H—HFS)
60–63		Start time of transmission
64–67		End time of transmission
68–71		Byte count of transmission
72		FTP ID (S—server)
73–75		Last reply sent to this client (FTP server)
76–119		For LOGIN records, this is the user ID of the failed login attempt. Otherwise, this is the data set name or up to the first 44 bytes of the HFS file name.
120–127		Member name for PDS
128–135		Reserved for abnormal end information
136–179		Second data set name, if needed (for example, Rename). For HFS, up to the first 44 bytes of the HFS file name.
180–187		Second member name, if needed (for example, Rename)
188–195		Started task qualifier
196–203		TCP/IP host name
204–205		Remote port number
206–207		Local port number
208–209		Offset to the first HFS file name field
210–211		Offset to the second HFS file name field

Two variable-length fields at the end of the record contain HFS file names. The variable-length HFS name fields have the following format:

Byte	Description
0–1	Length of the HFS file name.
3–x	HFS file name.

The value in the length field is the length of the HFS file name only; it does not include the two bytes for the length field itself. The maximum size of this variable field is 1025 bytes; the maximum value in the length subfield is 1023.

Appendix B. Related Protocol Specifications

This appendix lists the related protocol specifications for TCP/IP for MVS. The Internet suite of protocols 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 are so useful that they become recommended protocols. That is, all future implementations for TCP/IP are recommended to implement this particular function or protocol. These become the *de facto* standards, on which the TCP/IP protocol suite is built.

Many features of TCP/IP for MVS are based on the following RFCs:

RFC	Title and Author
768	*User Datagram Protocol* J.B. Postel
791	*Internet Protocol* J.B. Postel
792	*Internet Control Message Protocol* J.B. Postel
793	*Transmission Control Protocol* J.B. Postel
821	*Simple Mail Transfer Protocol* J.B. Postel
822	*Standard for the Format of ARPA Internet Text Messages* D. Crocker
823	*DARPA Internet Gateway* R.M. Hinden, A. Sheltzer
826	*Ethernet Address Resolution Protocol: or Converting Network Protocol Addresses to 48.Bit Ethernet Address for Transmission on Ethernet Hardware* D.C. Plummer
854	*Telnet Protocol Specification* J.B. Postel, J.K. Reynolds
855	*Telnet Option Specification* J.B. Postel, J.K. Reynolds
856	*Telnet Binary Transmission* J.B. Postel, J.K. Reynolds
857	*Telnet Echo Option* J.B. Postel, J.K. Reynolds
858	*Telnet Suppress Go Ahead Option* J.B. Postel, J.K. Reynolds
859	*Telnet Status Option* J.B. Postel, J.K. Reynolds
860	*Telnet Timing Mark Option* J.B. Postel, J.K. Reynolds
861	*Telnet Extended Options —List Option* J.B. Postel, J.K. Reynolds
862	*Echo Protocol* J.B. Postel
863	*Discard Protocol* J.B. Postel
864	*Character Generator Protocol* J.B. Postel
877	*Standard for the Transmission of IP Datagrams over Public Data Networks* J.T. Korb
885	*Telnet End of Record Option* J.B. Postel
903	*Reverse Address Resolution Protocol* R. Finlayson, T. Mann, J.C. Mogul, M. Theimer
904	*Exterior Gateway Protocol Formal Specification* D.L. Mills
919	*Broadcasting Internet Datagrams* J.C. Mogul

922 *Broadcasting Internet Datagrams in the Presence of Subnets* J.C. Mogul

950 *Internet Standard Subnetting Procedure* J.C. Mogul, J.B. Postel

952 *DoD Internet Host Table Specification* K. Harrenstien, M.K. Stahl, E.J. Feinler

959 *File Transfer Protocol* J.B. Postel, J.K. Reynolds

974 *Mail Routing and the Domain Name System* C. Partridge

1009 *Requirements for Internet Gateways* R.T. Braden, J.B. Postel

1013 *X Window System Protocol, Version 11: Alpha Update* R.W. Scheifler

1014 *XDR: External Data Representation Standard* Sun Microsystems Incorporated

1027 *Using ARP to Implement Transparent Subnet Gateways* S. Carl-Mitchell, J.S. Quarterman

1032 *Domain Administrators Guide* M.K. Stahl

1033 *Domain Administrators Operations Guide* M. Lottor

1034 *Domain Names—Concepts and Facilities* P.V. Mockapetris

1035 *Domain Names—Implementation and Specification* P.V. Mockapetris

1042 *Standard for the Transmission of IP Datagrams over IEEE 802 Networks* J.B. Postel, J.K. Reynolds

1044 *Internet Protocol on Network System's HYPERchannel: Protocol Specification* K. Hardwick, J. Lekashman

1055 *Nonstandard for Transmission of IP Datagrams over Serial Lines: SLIP* J.L. Romkey

1057 *RPC: Remote Procedure Call Protocol Version 2 Specification* Sun Microsystems Incorporated

1058 *Routing Information Protocol* C.L. Hedrick

1073 *Telnet Window Size Option* D. Waitzman

1079 *Telnet Terminal Speed Option* C.L. Hedrick

1091 *Telnet Terminal-Type Option* J. VanBokkelen

1094 *NFS: Network File System Protocol Specification* Sun Microsystems Incorporated

1096 *Telnet X Display Location Option* G. Marcy

1118 *Hitchhikers Guide to the Internet* E. Krol

1122 *Requirements for Internet Hosts—Communication Layers* R.T. Braden

1123 *Requirements for Internet Hosts—Application and Support* R.T. Braden

1155 *Structure and Identification of Management Information for TCP/IP-Based Internets* M.T. Rose, K. McCloghrie

1156 *Management Information Base for Network Management of TCP/IP-based Internets* K. McCloghrie, M.T. Rose

1157 *Simple Network Management Protocol (SNMP)* J.D. Case, M. Fedor, M.L. Schoffstall, C. Davin

1179 *Line Printer Daemon Protocol* The Wollongong Group, L. McLaughlin III

1180 *TCP/IP Tutorial* T.J. Socolofsky, C.J. Kale

1183 *New DNS RR Definitions* C.F. Everhart, L.A. Mamakos, R. Ullmann, P.V. Mockapetris, (Updates RFC 1034, RFC 1035)

1184 *Telnet Linemode Option* D. Borman

1187 *Bulk Table Retrieval with the SNMP* M.T. Rose, K. McCloghrie, J.R. Davin

1188 *Proposed Standard for the Transmission of IP Datagrams over FDDI Networks* D. Katz

1198 *FYI on the X Window System* R.W. Scheifler

1207 *FYI on Questions and Answers: Answers to Commonly Asked :q.Experienced Internet User:eq. Questions* G.S. Malkin, A.N. Marine, J.K. Reynolds

1208 *Glossary of Networking Terms* O.J. Jacobsen, D.C. Lynch

1213 *Management Information Base for Network Management of TCP/IP-Based Internets: MIB-II,* K. McCloghrie, M.T. Rose

1215 *Convention for Defining Traps for Use with the SNMP* M.T. Rose

1228 *SNMP-DPI Simple Network Management Protocol Distributed Program Interface* G.C. Carpenter, B. Wijnen

1229 *Extensions to the Generic-Interface MIB* K. McCloghrie

1230 *IEEE 802.4 Token Bus MIB IEEE 802 4 Token Bus MIB* K. McCloghrie, R. Fox

1231 *IEEE 802.5 Token Ring MIB IEEE 802.5 Token Ring MIB* K. McCloghrie, R. Fox, E. Decker

1267 *A Border Gateway Protocol 3 (BGP-3)* K. Lougheed, Y. Rekhter

1268 *Application of the Border Gateway Protocol in the Internet* Y. Rekhter, P. Gross

1269 *Definitions of Managed Objects for the Border Gateway Protocol (Version 3)* S. Willis, J. Burruss

1270 *SNMP Communications Services* F. Kastenholz, ed.

1340 *Assigned Numbers* J.K. Reynolds, J.B. Postel

1348 *DNS NSAP RRs* B. Manning

1350 *TFTP Protocol* K.R. Sollins

1351 *SNMP Administrative Model* J. Davin, J. Galvin, K. McCloghrie

1352 *SNMP Security Protocols* J. Galvin, K. McCloghrie, J. Davin

1353 *Definitions of Managed Objects for Administration of SNMP Parties* K. McCloghrie, J. Davin, J. Galvin

1354 *IP Forwarding Table MIB* F. Baker

1356 *Multiprotocol Interconnect on X.25 and ISDN in the Packet Mode* A. Malis, D. Robinson, R. Ullmann

1372 *Telnet Remote Flow Control Option* D. Borman, C. L. Hedrick

1374 *IP and ARP on HIPPI* J. Renwick, A. Nicholson

1381 *SNMP MIB Extension for X.25 LAPB* D. Throop, F. Baker

1382 *SNMP MIB Extension for the X.25 Packet Layer* D. Throop

1387 *RIP Version 2 Protocol Analysis* G. Malkin

1388 *RIP Version 2 — Carrying Additional Information* G. Malkin

1389 *RIP Version 2 MIB Extension* G. Malkin

1390 *Transmission of IP and ARP over FDDI Networks* D. Katz

1393 *Traceroute Using an IP Option* G. Malkin

1397 *Default Route Advertisement In BGP2 And BGP3 Versions of the Border Gateway Protocol* D. Haskin

1398 *Definitions of Managed Objects for the Ethernet-like Interface Types* F. Kastenholz

1540 *IAB Official Protocol Standards* J.B. Postel

1571 *Telnet Environment Option Interoperability Issues* D. Borman

1572 *Telnet Environment Option* S. Alexander

1592 *Simple Network Management Protocol Distributed Protocol Interface Version 2.0* B. Wijnen, G. Carpenter, K. Curran, A. Sehgal, G. Waters

1594 *FYI on Questions and Answers: Answers to Commonly Asked "New Internet User" Questions* A.N. Marine, J. Reynolds, G.S. Malkin

1695 *Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2*M. Ahmed, K. Tesink

1901 *Introduction to Community-Based SNMPv2* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1902 *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1903 *Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1904 *Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1905 *Protocols Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1906 *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1907 *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1908 *Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework* J. Case, K. McCloghrie, M. Rose, S. Waldbusser

1909 *An Administrative Infrastructure for SNMPv2* K. McCloghrie

1910 *User-based Security Model for SNMPv2* G. Waters

These documents can be obtained from:

Government Systems, Inc.
Attn: Network Information Center
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

Many RFCs are available online. Hard copies of all RFCs are available from the NIC, either individually or on a subscription basis. Online copies are available using FTP from the NIC at `nic.ddn.mil`. Use FTP to download the files, using the following format:

```
RFC:RFC-INDEX.TXT
RFC:RFCnnnn.TXT
RFC:RFCnnnn.PS
```

Where:

nnnn	Is the RFC number.
`TXT`	Is the text format.
`PS`	Is the PostScript format.

You can also request RFCs through electronic mail, from the automated NIC mail server, by sending a message to `service@nic.ddn.mil` with a subject line of `RFC` *nnnn* for text versions or a subject line of `RFC` *nnnn*`.PS` for PostScript versions. To request a copy of the RFC index, send a message with a subject line of `RFC INDEX`.

For more information, contact `nic@nic.ddn.mil`.

Appendix C. Description of Syslog Daemon (syslogd)

Syslog daemon (syslogd) is a server process that has to be started as one of the first processes in your OpenEdition environment. Other servers and stack components use syslogd for logging purposes and can also send trace information to syslogd.

Each application actives and deactivates traces in a slightly different manner. Refer to the chapter on the individual application for details. OS/390 TCP/IP OpenEdition components use the local1 and daemon facility names (see Figure 12 on page 261).

Servers on the local system use AF_UNIX sockets to communicate with syslogd; remote servers use the AF_INET socket. If syslogd is not started, application log data appears on the MVS console.

Format

syslogd [–**f** *conffile***] [**–**m** *markinterval***] [**–**p** *logpath***]**

Description

syslogd reads and logs system messages to the console, log files, other machines, or users as specified by the configuration file.

The configuration file is read at startup and whenever the hang-up signal (SIGHUP) is received. The syntax of the configuration file is described below.

syslogd stores its process id in file:

`/etc/syslog.pid`

so it may be used to terminate or re-configure the daemon.

Messages are read from the UNIX domain datagram socket and the Internet domain datagram socket. Kernel messages are not logged in OpenEdition MVS.

Options

syslogd recognizes the following options:

–**f** Configuration file name.

–**m** Number of minutes between mark messages.

–**p** Path name for the UNIX datagram socket.

Files

/dev/console Operator console.

/etc/syslog.pid Process id is written here.

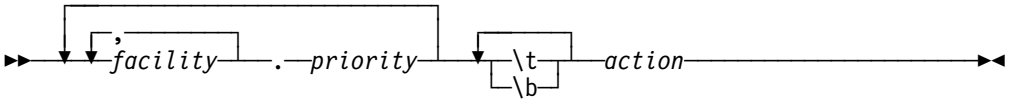
/etc/syslog.conf Default configuration file name.

/dev/log Default log path for UNIX datagram socket.

/usr/sbin/syslogd This is the server.

Configuration Lines

Each line of the configuration file has the following syntax:



where \t is a tab character and \b is a blank character.

The facilities supported are as follows:

kern Message generated by the system.

user Message generated by a process (user).

mail Message generated by mail system.

news Message generated by news system.

uucp Message generated by UUCP system.

daemon Message generated by system daemon.

auth/authpriv Message generated by authorization daemon.

cron Message generated by the clock daemon.

lpr Message generated by printer system.

local0 Reserved for local use.

local1 Reserved for local use.

local2 Reserved for local use.

local3 Reserved for local use.

local4 Reserved for local use.

local5 Reserved for local use.

local6 Reserved for local use.

local7 Reserved for local use.

mark Used for logging MARK messages.

The priorities supported are as follows:

emerg/panic A panic condition was reported to all processes.

alert A condition that should be corrected immediately.

crit A critical condition.

err(or) An error message.

warn(ing) A warning message.

notice A condition requiring special handling.

info A general information message.

debug A message useful for debugging programs.

none Do not log any messages for the facility.

The actions supported are as follows. Be sure to use lowercase for all filenames, users, and hosts:

/file log message to this file

@host log message to syslog daemon on another host

user1,user2,... log message to the list of users

***** log message to all logged-in users

Note: Comments can be added to the configuration file by placing the # character in column 1 of the comment line. Everything following the # character will be treated as a comment.

Syslog.conf Examples

Here is a sample of a syslog.conf file:

```
# Examples:
# log all daemon messages to the operator console.
# Note:  this may generate a lot of master console
# output if traces are currently active in several
# TCP/IP components
daemon.*              /dev/console
#
# All debug messages (and above priority
# messages) from telnet go to telnet.debug
local1.debug          /tmp/syslogd/telnet.debug
#
# All debug messages
# (and above priority messages) go to
# server.debug
daemon.debug          /tmp/syslogd/server.debug
#
# log mail messages at info and above to /tmp/user.info
mail.info /tmp/user.info
#
#
#user1 and user2 should get all emergency messages
*.alert user1,user2
#
# log all messages (except mail) to /tmp/all.except.mail
*.*;mail.none /tmp/all.except.mail
#
# log clock and printer err(+) messages to yourhost
#
cron,lpr.err @yourhost
```

Figure 12. Example of a syslog.conf File

Starting Syslogd

If you want ITRACE messages from TCP/IP initialization written to syslogd and do not want trace messages from TCP/IP or inetd written to the master console, you must start syslogd before TCP/IP and inetd:

1. Start syslogd
2. Start TCP/IP
3. Start inted

For special considerations on remote syslogd servers, see “Usage Notes.”

Usage Notes

- **syslogd** can only be started by a superuser.
- **syslogd** can be terminated using the SIGTERM signal.
- If you want syslod in an MVS image to receive log data from remote syslogd servers, then UDP port 514 must be reserved for OMVS in your OS/390 TCP/IP OpenEdition PROFILE data set.

```
PORT
   . . .
    514 UDP OMVS              ; OE SyslogD Server
   . . .
```

- If there is no TCP/IP (AF_INET) transport connected to OE when syslogd starts, syslogd cannot bind to port 514, so it cannot receive log data from remote syslogd servers. If you want data from remove syslogd servers, you must stop and restart syslogd after TCP/IP has been initialized.

  Note: If trace messages were being written at the time syslogd was stopped and restarted, all subsequent trace messages will be lost.

- If the configuration file cannot be opened, the following configuration lines are used as a default:

```
*.err    /dev/console
*.panic  *
```

- Configuration file errors are written to the operator console because initialization is not complete until the entire configuration file has been read.
- Facility mark is not affected by the *.priority usage.
- Minimum mark interval is 30 seconds.

Exit Values

If an invalid argument is entered, or if an invalid number of arguments is entered, then a return code of 1 is set and syslogd exits. All other cases in which syslogd exits cause a return code of zero to be set.

Related Information

To terminate (“kill”) syslogd, send a SIGTERM (terminate). A SIGHUP (hangup) will cause it to reread its configuration file. To send a signal to syslogd, issue one of the following:

- `KILL -s SIGHUP <PID>` or
- `KILL -s SIGTERM <PID>`

You can obtain the PID by reading the PID file that syslogd opens in the /etc directory.

For more information about syslogd, refer to *Accessing OS/390 OpenEdition MVS from the Internet* (SG24-4721).

Appendix D. Setting up the inetd Configuration File

inetd is a generic listener program used by such servers as OE TELNETD and OE REXECD. Other servers such as OE FTPD have their own listener program and do not used inetd.

inetd.conf is an example of the user's configuration file. It is stored in the /etc directory. Upon startup, the OE TELNETD server, rshell, rlogin, and rexec are initiated. If it does not include OE TCP/IP applications, add the following:

```
#======================================================================
# service | socket | protocol | wait/ | user | server  | server program
# name    | type   |          | nowait|      | program |   arguments
#======================================================================
#
 shell    stream    tcp        nowait  OMVSKERN /usr/sbin/orshd rshd -l
 exec     stream    tcp        nowait  OMVSKERN /usr/sbin/orexecd rexecd -LV
 otelnet  stream    tcp        nowait  OMVXKERN /usr/sbin/otelnetd otelnetd  -LV
```

Figure 13. Adding Applications to /etc/inetd.conf

To establish a relationship between the servers defined in the /etc/inetd.conf file and specific port numbers in the OpenEdition environment, insure that statements have been added to ETC.SERVICES for each of these servers. See the sample ETC.SERVICES installed in the /usr/lpp/tcpip/samples/services directory for how to specify ETC.SERVICES statements for these servers.

The traces for both the OE REXECD server and the OE RSHD server are enabled via options in the inetd configuration file (/etc/inetd.conf):

```
#======================================================================
# service | socket | protocol | wait/ | user | server  | server program
# name    | type   |          | nowait|      | program |   arguments
#======================================================================
#
shell      stream    tcp       nowait OMVSKERN /usr/sbin/orshd rshd -d  1
exec       stream    tcp       nowait OMVSKERN /usr/sbin/orexecd rexecd -d  2
```

Figure 14. Setting Traces in /etc/inetd.conf

The traces are turned on for both servers by passing a -d argument to the server programs. **1** is the RSHD server and **2** is the REXECD server. All commands executed after the debug flags have been turned on in the inetd configuration file and the inetd server has reread the file will produce trace output.

The trace is written in formatted form to the syslogd facility name daemon with a priority of debug. The trace data can be routed to a file in your Hierarchical File System by specifying the following definition in your syslogd configuration file (/etc/syslogd.conf):

```
  #
  # All ftp, rexecd, rshd
  # debug messages (and above
  # priority messages) go
# to server.debug.a
#
daemon.debug              /tmp/syslogd/server.debug.a
```

In this example, the trace data is written to /tmp/syslogd/daemon.debug.a in your Hierarchical File System. For more information on syslogd, refer to Appendix C, “Description of Syslog Daemon (syslogd)” on page 259.

For more information about inetd, refer to *OS/390 OpenEdition Planning*, SC28-1890-02 or *Accessing OS/390 OpenEdition MVS from the Internet* (SG24-4721).

Appendix E. How to Read a Syntax Diagram

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.

Parameters

The following types of parameters are used in syntax diagrams.

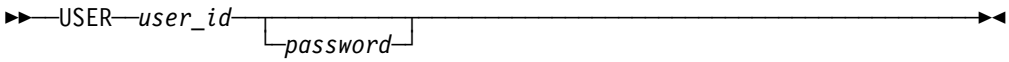
Parameter	Description
Required	Required parameters are displayed on the main path.
Optional	Optional parameters are displayed below the main path.
Default	Default parameters are displayed above the main path.

Parameters are classified as keywords or variables. Keywords are displayed in uppercase letters and can be entered in uppercase or lowercase. For example, a command name is a keyword.

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

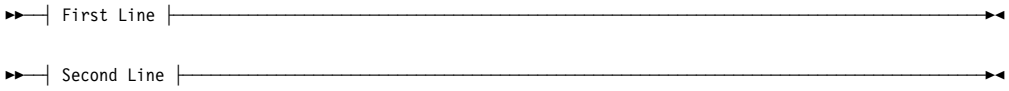
Syntax Examples

In the following example, the USER command is a keyword. The required variable parameter is *user_id*, and the optional variable parameter is *password*. Replace the variable parameters with your own values.



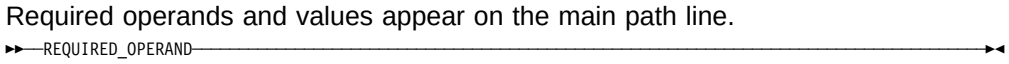
Longer than one line

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.



Required operands

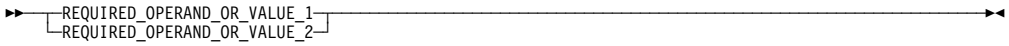
Required operands and values appear on the main path line.



You must code required operands and values.

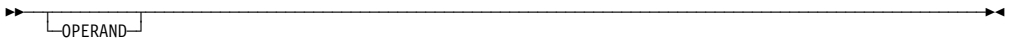
Choose one required item from a stack

If there is more than one mutually exclusive required operand or value to choose from, they are stacked vertically in alphanumeric order.



Optional values

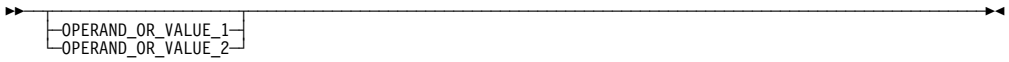
Optional operands and values appear below the main path line.



You can choose not to code optional operands and values.

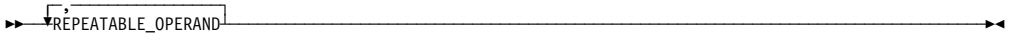
Choose one optional operand from a stack

If there is more than one mutually exclusive optional operand or value to choose from, they are stacked vertically in alphanumeric order below the main path line.



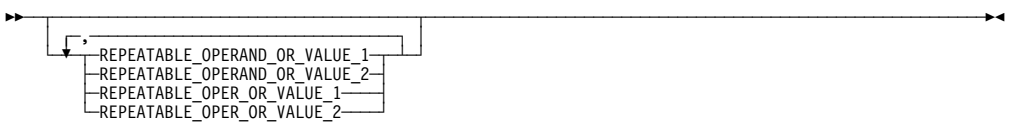
Repeating an operand

An arrow returning to the left above an operand or value on the main path line means that the operand or value can be repeated. The comma means that each operand or value must be separated from the next by a comma.



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.



If an operand or value can be abbreviated, the abbreviation is described in the text associated with the syntax diagram.

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

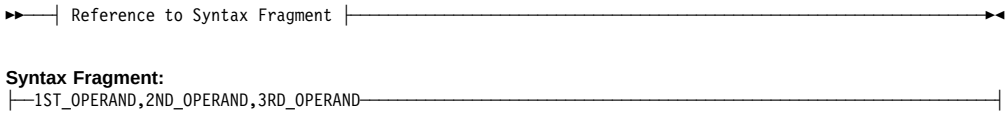
Default operands and values appear above the main path line. TCP/IP uses the default if you omit the operand entirely.

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.



Appendix F. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

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

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:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
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.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

IBM is required to include the following statements in order to distribute portions of this document and the software described herein to which contributions have been made by The University of California.

Portions herein © Copyright 1979, 1980, 1983, 1986, Regents of the University of California. Reproduced by permission. Portions herein were developed at the Electrical Engineering and Computer Sciences Department at the Berkeley campus of the University of California under the auspices of the Regents of the University of California.

Trademarks

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

ACF/VTAM
AD/Cycle
AIX
AIX/ESA
BookManager
C/370
CICS
DB2
DFSMS
DFSMS/MVS
ESCON
ES/9000
EtherStreamer
Extended Services
GDDM
Hardware Configuration Definition
IBM
LANStreamer
Library Reader
MVS/ESA
MVS/SP
MVS/XA
NetView
OpenEdition
OS/2
OS/390
PS/2
RACF
RISC System/6000
RS/6000
SAA
System/370
System/390
VTAM
3090

The following terms are trademarks of other companies:

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

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

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

Glossary

The IBM Networking Software Glossary is now available in HTML format as well as PDF. You can access it directly at the following URL:

`http://www.networking.ibm.com/nsg/nsggls.htm`

This glossary includes terms and definitions from:

- The *American National Standards Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The ANSI/EIA Standard—440-A, *Fiber Optic Terminology* Copies may be purchased from the Electronic Industries Association, 2001 Pennsylvania Avenue, N.W., Washington, DC 20006. Definitions are identified by the symbol (E) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.
- Internet Request for Comments: 1208, *Glossary of Networking Terms*
- Internet Request for Comments: 1392, *Internet Users' Glossary*
- The *Object-Oriented Interface Design: IBM Common User Access Guidelines*, Carmel, Indiana: Que, 1992.

Bibliography

This bibliography lists the publications for IBM TCP/IP products.

IBM TCP/IP Publications

The following sections describe the books associated with IBM TCP/IP products.

OS/390 TCP/IP OpenEdition Publications

- *OS/390 TCP/IP OpenEdition Configuration Guide*, SC31-8304-00.

  This book is for people who want to configure, customize, administer, and maintain OS/390 TCP/IP OpenEdition. Familiarity with MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *OS/390 TCP/IP OpenEdition Diagnosis Guide*, SC31-8492-00.

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the OS/390 TCP/IP OpenEdition product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *OS/390 TCP/IP OpenEdition Messages and Codes*, SC31-8307-00.

  This book explains the informational and error messages issued by OS/390 TCP/IP OpenEdition. It can help users, operators, or system programmers to diagnose and fix problems identified by error messages.

- *OS/390 TCP/IP OpenEdition Planning and Release Guide*, SC31-8303-00.

  This book is intended to help you plan for OS/390 TCP/IP OpenEdition whether you are migrating from a previous version or installing TCP/IP for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with OS/390 TCP/IP OpenEdition.

- *OS/390 TCP/IP OpenEdition Programmer's Reference*, SC31-8308-00

  This book 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.

  This book is for people who want to use the supplied interfaces while writing application programs that access OS/390 TCP/IP OpenEdition. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *OS/390 TCP/IP OpenEdition User's Guide*, GC31-8305-00.

  This book is for people who want to use OS/390 TCP/IP OpenEdition for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for MVS Publications

- *TCP/IP Version 3 for OpenEdition MVS: Applications Feature Guide*, SC31-8069-00.

  This book explains how to plan for, install, customize, and use the OpenEdition MVS Applications Feature. The Feature consists of applications and interfaces for direct access to the OpenEdition MVS environment. For example, users of the Feature can use MVS, UNIX, or AIX commands to transfer files, log in to the OpenEdition environment without going through TSO, and run commands remotely. This book also explains how to improve performance and diagnose problems when using the Feature.

- *TCP/IP for MVS: Application Programming Interface Reference*, SC31-7187-02.

  This book 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 book to adapt your existing applications to communicate with each other using sockets over TCP/IP.

- *TCP/IP for MVS: CICS TCP/IP Socket Interface Guide and Reference*, SC31-7131-02.

  This book is for people who want to set up, write application programs for, and diagnose problems with the socket interface for CICS using TCP/IP for MVS.

- *TCP/IP for MVS: Customization and Administration Guide*, SC31-7134-03.

  This book is for people who want to con customize, administer, and maintain TCP/IP for MVS. Familiarity with MVS operating system, TCP/IP protocols,

and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: Diagnosis Guide*, LY43-0105-02.

  This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the IBM TCP/IP for MVS product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

- *TCP/IP for MVS: IMS TCP/IP Application Development Guide and Reference*, SC31-7186-02.

  This book is for programmers who want application programs that use the IMS TCP/IP application development services provided by IBM TCP/IP for MVS.

- *TCP/IP for MVS: Messages and Codes*, SC31-7132-03.

  This book explains the informational a nd error messages issued by IBM TCP/IP for MVS. It can help users, operators, or system programmers to diagnose and fix problems identified by TCP/IP for MVS error messages.

- *TCP/IP for MVS: Network Print Facility*, SC31-8074-03.

  This book is for system programmers an d network administrators who need to prepare their network to route VTAM, JES2, or JES3 printer output to remote printers using TCP/IP for MVS.

- *TCP/IP for MVS: Offloading TCP/IP Processing*, SC31-7133-02.

  This book is for people who want to install and configure the Offload feature on IBM 3172 Model 3 Interconnect Controllers. This book is also for people who want to use and customize the Offload feature of TCP/IP for MVS.

- *TCP/IP for MVS: Planning and Migration Guide*, SC31-7189-01.

  This book is intended to help you plan for TCP/IP for MVS whether you are migrating from a previous version or installing TCP/IP for MVS for the first time. This book also identifies the suggested and required modifications needed to enable you to use the enhanced functions provided with TCP/IP for MVS.

- *TCP/IP: Performance Tuning Guide*, SC31-7188-02.

  This book describes how to improve the performance of your network operations.

- *TCP/IP for MVS: Programmer's Reference*, SC31-7135-02.

  This book 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.

  This book is for people who want to use the supplied interfaces while writing application programs that access TCP/IP for MVS. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

- *TCP/IP for MVS: User's Guide*, SC31-7136-02.

  This book is for people who want to use TCP/IP for MVS for data communication. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

TCP/IP for VM Publications

The following list describes books in the IBM TCP/IP for VM library.

- *IBM TCP/IP Version 2 Release 4 for VM: Messages and Codes*, SC31-6151-03.

  This book is for system programmers who want to diagnose and fix problems identified by TCP/IP for VM error messages.

- *IBM TCP/IP Version 2 Release 4 for VM: Planning and Customization*, SC31-6082-03.

  This book is for system programmers who want to plan and customize the TCP/IP for VM environment.

- *IBM TCP/IP Version 2 Release 4 for VM: Programmer's Reference*, SC31-6084-03.

  This book is for application and system programmers who want to write application programs that use TCP/IP for VM. Application programmers should know the VM operating system.

- *IBM TCP/IP Version 2 Release 4 for VM: User's Guide*, SC31-6081-03.

  This book is for people who want to use TCP/IP for VM for data communication. Familiarity with VM operating system, IBM Command Processor (CP), and IBM Conversational Monitor System (CMS) is recommended.

TCP/IP for OS/2 Publication

IBM TCP/IP Version 3.0 for OS/2: Programmer's Reference, SC31-6077.

This book provides application and system programmers with the information required to write application programs that use TCP/IP for OS/2. Programmers should know the OS/2 operating system.

TCP/IP for DOS Publications

The following list describes books in the IBM TCP/IP for DOS library.

- *IBM TCP/IP Version 2.1.1 for DOS: Command Reference*, SX75-0083.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, understand DOS operating system concepts, and be familiar with the *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*

- *IBM TCP/IP Version 2.1.1 for DOS: Installation and Administration*, SC31-7047.

  This book provides system programmers, network administrars, and workstation users responsible for installing TCP/IP for DOS with the information required to plan and implement the installation of TCP/IP for DOS. The topics include hardware and software requirements, pre-installation system performance considerations, instructions for installing TCP/IP for DOS, instructions for customizing the TCP/IP for DOS environment, and installation examples.

- *IBM TCP/IP Version 2.1.1 for DOS: Programmer's Reference*, SC31-7046.

  This book is for application and system programmers to aid them in writing application programs that use TCP/IP for DOS on a workstation. Application programmers should know the DOS operating system and multitasking operating system concepts. Application programmers should be knowledgeable in the C programming language.

- *IBM TCP/IP Version 2.1.1 for DOS: User's Guide*, SC31-745.

  This book is for people who use a workstation with TCP/IP for DOS, such as end users and system programmers. The people who use this book should be familiar with DOS and the workstation, and also understand DOS operating system concepts.

TCP/IP for AIX (RS/6001, PS/2, RT, 370) Publications

The following list shows books in the TCP/IP for AIX library.

- *AIX Operating System TCP/IP User's Guide*, SC23-2309.
- *AIX PS/2 TCP/IP User's Guide*, SC23-2047.
- *TCP/IP for IBM X-Windows on DOS 2.1*, SC23-2349.

TCP/IP for AS/400 Publications

The following list shows books in the TCP/IP for AS/400 library.

- *IBM AS/400 Communications: TCP/IP Guide*, SC41-9875.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.

Other IBM TCP/IP Publications

The following list shows other available IBM TCP/IP books.

- *IBM Local Area Network Technical Reference*, SC30-3383.
- *IBM TCP/IP for VM and MVS: Diagnosis Guide*, LY43-0013.
- *TCP/IP and National Language Support*, GG24-3840.
- *TCP/IP Introduction*, GC31-6080.
- *TCP/IP Tutorial and Technical Overview*, GG24-3376.

IBM Operating System Publications

The following lists show books about various IBM operating systems.

AIX Publications

- *AIX Communications Concepts and Procedures for IBM RISC System/6001*, GC23-2203.
- *AIX Communications Programming Concepts*, SC23-2206.
- *IBM AIX Operating System Technical Reference, Volume 1*, SC23-2300.
- *IBM AIX Operating System Technical Reference, Volume 2*, SC23-2301.

AS/400 Publications

- *IBM AS/400 CL Reference Manual Volume 1*, SC21-9775.
- *IBM AS/400 CL Reference Manual Volume 2*, SC21-9776.
- *IBM AS/400 CL Reference Manual Volume 3*, SC21-9777.
- *IBM AS/400 CL Reference Manual Volume 4*, SC21-9778.

- *IBM AS/400 CL Reference Manual Volume 5*, SC21-9779.
- *IBM AS/400 Communications: APPN Network User's Guide*, SC21-8188.
- *IBM AS/400 Communications: Programmer's Guide*, SC21-9590.
- *IBM AS/400 Communications: User's Guide*, SC21-9601.
- *IBM AS/400 Device Configuration Guide*, SC21-8106.
- *IBM AS/400 Programming: Command Reference Summary*, SC21-8076.
- *IBM AS/400 Programming: Data Management Guide*, SC21-9658.
- *IBM AS/400 System Operations: Database Coordinator' Guide*, SC21-8086.
- *IBM AS/400 System Operations: Operator's Guide*, SC21-8082.

DOS Publications

- *DOS Getting Started Version 5.00*, SA40-0637.
- *DOS 5.02 Technical Reference*, S16G-4559.
- *DOS/Windows Client Getting Started*, SC09-3001.
- *PC DOS 6.1 Command Reference*, S71G-3634.

MVS Publications

For a complete description of the library for MVS/ESA Version 5, see *OS/390 Information Roadmap*, GC28-1727-02. See also “JES Publications” on page 280.

OS/2 Publications

- *IBM OS/2 Warp Server Up and Running!*, S25H-8004
- *IBM Official Guide to Using OS/2 Warp*, ISBN 1-56884-466-2 (Karla Stagray and Linda S. Rogers; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)
- *IBM OS/2 Warp Internet Connection: Your Key to Cruising the Internet and the World Wide Web*, ISBN 1-56884-465-4 (Deborah Morrison; Foster City, CA: An IBM Press Book published by IDG Books Worldwide, Inc., 1995)

OS/390 Publications

- *OS/390 Information Roadmap*, GC28-1727-02

  This book describes the documentation for the specific elements included in OS/390.
- *OS/390 Planning for Installation Release 3*, GC28-1726-02

  This book is intended to help you plan for the installation of OS/390. It describes migration, installation, hardware and software requirements, and coexistence considerations.
- *OS/390 OpenEdition Introduction*, GC28-1889-01.
- *OS/390 OpenEdition Planning*, SC28-1890-02.
- *OS/390 OpenEdition User's Guide*, SC28-1891-02.
- *OS/390 OpenEdition Command Reference*, SC28-1892-02.
- *OS/390 OpenEdition Messages and Codes*, SC28-1908-02.
- *OS/390 Language Environment Programming Guide*, SC28-1939-02.
- *OS/390 Language Environment Programming Reference*, SC28-1940-02.
- *OS/390 OpenEdition Programming: Assembler Callable Services Reference*, SC28-1899-02.
- *OS/390 Open Systems Adapter Support Facility Users's Guide*, SC28-1855.
- *Planning for the System/390 Open Systems Adapter Feature*, GC23-3870.

VM Publications

- *VM/ESA CMS Command Reference Summary*, SX24-5249.
- *VM/ESA CP Planning and Administration for 370*, SC24-5430.
- *VM/ESA CP Programming Services for 370*, SC24-5435.
- *VM/ESA Group Control System Reference for 370*, SC24-5426.
- *VM/ESA: Library Guide and Master Index*, GC23-0367.
- *VM/ESA: Master Index for 370*, GC24-5436.
- *VM/ESA Service Introduction and Reference*, SC24-5444.
- *VM/SP CMS Command Reference*, ST00-1981.
- *VM/SP Group Control System Macro Reference*, SC24-5250.
- *VM/SP Installation Guide*, SC24-5237.
- *VM/SP High Performance Option:*

Library Guide and Master Index, GC23-0187.

- *VM/SP System Facilities for Programming*, SC24-5288.
- *VM/XA CP Programming Services*, SC23-0370.
- *VM/XA Diagnosis Reference*, LY27-8054.
- *VM/XA Installation and Service*, SC23-0364.
- *VM/XA SP Group Control System Command and Macro Reference*, SC23-0433.

IBM Software Publications

The following sections describe the books associated with IBM software products.

ACF/VTAM Publications

The following list shows books in the VTAM Version 4 Release 4 library.

- *VTAM Installation and Migration Guide*, GC31-8367-00.
- *VTAM Release Guide*, GC31-6545-00.
- *VTAM Network Implementation Guide*, SC31-8370-00.
- *VTAM Resource Definition Reference*, SC31-8377-00.
- *VTAM Resource Definition Samples*, SC31-8378-00.
- *VTAM Customization*, LY43-0075-00.
- *VTAM Operation*, SC31-8372-00.
- *VTAM Messages*, GC31-8368-00.
- *VTAM Codes*, GC31-8369-00.
- *VTAM Programming*, SC31-8373-00.
- *VTAM Guide to Programming for LU 6.2*, SC31-8374-00.
- *VTAM Programming Reference for LU 6.2*, SC31-8375-00.
- *VTAM Programming for CSM*, SC31-8420-00.
- *VTAM CMIP Services and Topology Agent Programming Guide*, SC31-8365-00.
- *VTAM Diagnosis*, LY43-0078-00.
- *VTAM Data Areas for MVS/ESA Volume 1*, LY43-0076-00.
- *VTAM Data Areas for MVS/ESA Volume 2*, LY40-0077-00.
- *APPC Application Suite User's Guide*, SC31-6532-00.
- *APPC Application Suite Administration*, SC31-6533-00.
- *APPC Application Suite Programming*, SC31-6534-00.
- *VTAM AnyNet Guide to Sockets over SNA*, SC31-8371-00.
- *VTAM AnyNet Guide to SNA over TCP/IP*, SC31-8376-00.
- *VTAM Glossary*, GC31-8366-00.
- *Planning for NetView, NCP, and VTAM*, SC31-8063-00.
- *Planning for Integrated Networks*, SC31-8062-00.
- *VTAM Licensed Program Specifications*, GC31-8379-00.
- *VTAM Operation Quick Reference*, SX75-0208-00.

DATABASE 2 Publications

The following lists show books in the DATABASE 2 library.

DATABASE 2 Version 2

- *IBM DATABASE 2 Version 2: Administration Guide*, SC26-4374.
- *IBM DATABASE 2 Version 2: Application Programming and SQL Guide*, SC26-4377.
- *IBM DATABASE 2 Version 2: Messages and Codes*, SC26-4379.
- *IBM DATABASE 2 Version 2: Reference Summary*, SX26-3771.
- *IBM DATABASE 2 Version 2: SQL Reference*, SC26-4380.

DATABASE 2 Version 3

- *IBM DATABASE 2 Version 3: DB2 Administration Guide*, SC26-4888.
- *IBM DATABASE 2 Version 3: DB2 Application Programming and SQL Guide*, SC26-4889.
- *IBM DATABASE 2 Version 3: DB2 Messages and Codes*, SC26-4892.
- *IBM DATABASE 2 Version 3: DB2 Reference Summary*, SX26-3801.
- *IBM DATABASE 2 Version 3: DB2 SQL Reference*, SC26-4890.

ISPF Publication

ISPF Dialog Management Guide and Reference, SC34-4266.

JES Publications

- *MVS/ESA Library Guide with JES2*, GC28-1423.
- *MVS/ESA Library Guide with JES3*, SC28-1424

MVS/DFP Publications

- *MVS/DFP Version 3 Release 3: Customizing and Operating the Network File System Server*, SC26-4832.
- *MVS/DFP Version 3 Release 3: Macro Instructions for Data Sets*, S26-4747.
- *MVS/DFP Version 3 Release 3: Using Data Sets*, SC26-4749.
- *MVS/DFP Version 3 Release 3: Using the Network File System Server*, SC26-4732.

Network Control Program (NCP) Publications

- *ACF/NCP V7R1 IP Router Planning and Installation Guide*, GG24-3974.
- *NCP and EP Reference*, LY43-0029.
- *NCP, SSP, and EP Generation and Loading Guide*, SC31-6221.
- *NCP, SSP, and EP Resource Definition Guide*, SC31-6223.
- *NCP, SSP, and EP Resource Definition Reference*, SC31-6224.

TME 10 NetView for OS/390 Publications

For a complete description of the TME 10 NetView for OS/390 library, see the *TME 10 NetView for OS/390 Library Reference*, SC31-8249.

Networking Systems Cross-Product Library

The following list shows books in the Networking Systems cross-product library.

- *Planning Aids: Pre-Installation Planning Checklist for NetView, NCP, and VTAM*, SX75-0092.
- *Planning for Integrated Networks*, SC31-8062.
- *Planning for NetView, NCP, and VTAM*, SC31-8063.

OpenEdition MVS Publications

The following list shows selected books in the OpenEdition MVS library.

- *OS/390 OpenEdition Introduction*, GC28-1889-01
- *OS/390 OpenEdition Planning*, SC28-1890-02

Programming Publications

The following list shows books about various programming applications.

- *IBM C/370 Diagnosis Guide and Reference* LY09-1804 (feature 8082).
- *IBM C/370 General Information Manual* GC09-1386.
- *IBM C/370 Installation and Customization Guide Version 2 Release 1.0*, GC09-1387.
- *IBM C/370 Programming Guide*, SC09-1384.
- *IBM C/370 Reference Summary*, SX09-1211.
- *IBM C/370 User's Guide*, SC09-1264.
- *OS/390 C/C++ Run-Time Library Reference*, SC28-1663-01.
- *IBM TSO Extensions CLISTs*, SC28-1876.
- *IBM TSO Extensions Command Language Reference* GX23-0015.
- *IBM TSO Extensions Interactive Data Transmission Facility: User's Guide*, SC28-1104.
- *IMS/ESA V3R1 Application Programming: DL/I Calls* SC26-4274.
- *HiPPI User's Guide and Programmer's Reference*, SA23-0369.
- *Parallel I/O Access Methods Programmer's Guide*, SC26-4648.
- *VS Pascal Application Programming Guide* SC26-4319.
- *VS Pascal Diagnosis Guide and Reference* LY27-9525.
- *VS Pascal General Information*, GT00-2664.
- *VS Pascal Installation and Customization for MVS* SC26-4321.
- *VS Pascal Installation and Customization for VM* SC26-4342.
- *VS Pascal Language Reference*, SC26-4320.

RACF Publications

The following list shows books in the RACF library.

- *IBM Resource Access Control Facility (RACF): General Information Manual*, GT00-2820.
- *IBM Resource Access Control Facility (RACF): User's Guide*, SC28-1341.
- *External Security Interface (RACROUTE) Macro Reference*, GC28-1366.
- *RACF Publications Order Guide*, GX22-0012.
- *Resource Access Control Facility (RACF) Security Administrator's Guide*, SC28-1340.
- *System Programming Library: RACF*, SC28-1343.

SMP/E Publications

The following list shows books in the SMP/E Release 8 library.

- *SMP/E Diagnosis Guide*, SC23-3130.
- *SMP/E Messages and Codes*, SC28-1107.
- *SMP/E Reference*, SC28-1107.
- *SMP/E Reference Summary*, SX22-0016.
- *SMP/E User's Guide*, SC28-1302.

VSAM Publication

MVS/370 VSAM Administration Guide, GC26-4066.

X.25 NPSI Publications

The following list shows books in the X.25 NPSI library.

- *X.25 Network Control Program Packet Switching Interface Diagnosis, Customization, and Tuning Version 3*, LY30-5610.
- *X.25 Network Control Program Packet Switching Interface Host Programming*, SC30-3502.
- *X.25 Network Control Program Packet Switching Interface Planning and Installation*, SC30-3470.

IBM Hardware Publications

The following sections describe the books associated with IBM hardware products.

System/370 and System/390 Publications

The following list shows the principles of operation manuals for the System/370 and System/390 processors.

- *IBM ESA/370 Principles of Operation*, SA22-7200.
- *IBM ESA/390 Principles of Operation*, SA22-7201.
- *IBM System/370 Extended Architecture Principles of Operation*, SA22-7085.
- *IBM System/370 Principles of Operation*, GA22-7001.
- *S/360, S/370, and S/390 I/O Interface Channel to Channel Control Unit OEMI*, GA22-6974.

3172 Interconnect Controller Publications

The following list shows books in the IBM 3172 Interconnect Controller library.

- *IBM Interconnect Controller Program User's Guide*, SC30-3525.
- *IBM 3172 Interconnect Controller Installation and Service Guide*, GA27-3861.
- *IBM 3172 Interconnect Controller Operator's Guide*, GA27-3860.
- *IBM 3172 Interconnect Controller Planning Guide*, GA27-3867.
- *IBM 3172 Interconnect Controller Status Codes*, GA27-3951.

3270 Information Display System Publication

3270 Information Display System: 3270 Data Stream Programmer's Reference, GA23-0059.

8232 LAN Channel Station Publications

The following list shows books in the IBM 8232 LAN Channel Station library.

- *IBM LAN Channel Support Program: Version 1.0 User's Guide*, SC30-3458.
- *IBM 8232 LAN Channel Station: Installation and Testing*, GA27-3796.
- *IBM 8232 LAN Channel Station: Operating Guide*, GA27-3785.

9370 Publications

The following list shows books in the 9370 library.

- *IBM 9370 Information System: Using the X.25 Communications Subsystem*, SA09-1742.
- *IBM 9370 Information System X.25 Communications Subsystem Description*, SA09-1743.
- *VM/ESA: Connectivity Planning, Administration, and Operation Release 1*, SC24-5448.

Other TCP/IP-Related Publications

The following sections describe other books associated with TCP/IP.

- *The Art of Distributed Application: Programming Techniques for Remote Procedure Calls* John R. Corbin, Springer-Verlog, 1991.
- *CAE Specification: X/Open Transport Interface (XTI),* X/Open Company Ltd., U. K., 1992, SC31-8005.
- *IEEE Network Magazine*, July 1990.
- *TCP/IP Illustrated Volume I: The Protocols*, W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1994, SR28-5586.
- *TCP/IP Illustrated Volume II: The Implementation*, Gary R. Wright and Richard Stevens, Addison-Wesley Publishing Company, Inc., 1995, SR28-5630.
- *TCP/IP Illustrated Volume III*, W. Richard Stevens, Addison-Wesley Publishing Company, Inc., 1996, SR23-7289
- *Interoperability Report*, Volume 3, No. 3, March 1989.
- "MIB II Extends SNMP Interoperability," C. Vanderberg, *Data Communications*, October 1990.
- *"Network Management and the Design of SNMP,"* J.D. Case, J.R. Davin, M.S. Fedor, M.L. Schoffstall.
- *"Network Management of TCP/IP Networks: Present and Future,"* A. Ben-Artzi, A. Chandna, V. Warrier.
- *The Simple Book: An Introduction to Management of TCP/IP-based Internets*, Marshall T Rose, Prentice Hall, Englewood Cliffs, New Jersey,1993.
- "Special Issue: Network Management and Network Security," *ConneXions-The Interoperability Report* Volume 4, No. 8, August 1990.
- *UNIX Programmer's Reference Manual* (4.3 Berkeley Software Distribution, Virtual VAX-11 Version). Department of Electrical Engineering and Computer Science. University of California, Berkeley, 1988.

OSF/Motif Publications

The following list shows OSF/Motif books.

- *OSF/Motif Application Environment Specifications* (AES), Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640483-9.
- *OSF/Motif Programmer's Guide* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640509-6.
- *OSF/Motif Programmer's Reference* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640517-7.
- *OSF/Motif Style Guide* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640491-X.
- *OSF/Motif User's Guide* Open Software Foundation, Prentice Hall, Inc., 1990, ISBN 0-13-640525-8.

Sun (RPC) Publications

The following list shows Sun Microsystems books.

- *Networking on the Sun Workstation: Remote Procedure Call Programming Guide* (800-1324-03), Sun Microsystems, Inc.
- *Network Programming* (800-1779-10), Sun Microsystems, Inc.

X Window System Publications

The following list shows X Window System books.

- *Introduction to the X Window System*, Oliver Jones, Prentice-Hall, 1988, ISBN 0-13-499997-5.
- *PEXlib Specification and C Language Binding* Jeff Stevenson, Hewlett-Packard Company, 1992, SR28-5116.
- *The X Window System Series* (6 volumes), O'Reilly & Associates, 1988, 1989, 1990, ISBN 0-937175-40-4, 0-937175-27-7, 0-937175-28-5, 0-937175-35-6, 0-937175-33-1, 0-937175-35-8.
- *X Protocol Reference Manual* Adrian Nye, ed. O'Reilly & Associates, Inc., 1990, ISBN 0-937175-50-1.
- *X Window System: C Library and Protocol Reference* Robert Scheifler, James Gettys, and Ron Newman, DEC Press, 1988, ISBN 1-55558-012-2.
- *X Window System: Programming and Applications with Xt*, Douglas A. Young, Prentice-Hall, 1989, ISBN 0-13-972167-3.

- *X Window System: Programming and Applications with Xt, OSF/Motif Edition* Douglas A. Young, Prentice-Hall, 1990, ISBN 0-13-497074-8.
- *X Window System Technical Reference*, Steven Mikes, Addison-Wesley, 1990, ISBN 0-201-52370-1.
- *X Window System User's Guide* Valerie Quercia and Tim O'Reilly, O'Reilly & Associates, Inc., 1990, ISBN 0-937175-14-5.

Network Architecture Publications

The following sections list books associated with network architecture.

Open Systems Interconnection (OSI) Publication

Open Systems Interconnection, Z320-9757.

Systems Network Architecture (SNA) Publications

The following list shows books in the SNA library.

- *Systems Network Architecture: Sessions between Logical Units*, GC20-1868.
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic*, SC30-3112.
- *Systems Network Architecture Format and Protocol Reference Manual: Management Services*, SC30-3346.
- *Systems Network Architecture Formats* GA27-3136.
- *Systems Network Architecture Network Product Formats*, LY43-0081.

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

Communicating Your Comments to IBM

OS/390 TCP/IP OpenEdition
Configuration Guide

Publication No. SC31-8304-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments by FAX, use this number:

  United States and Canada: **1-800-227-5088**
- If you prefer to send comments electronically, use this network ID:
  - IBM Mail Exchange: **USIB2HPD at IBMMAIL**
  - IBMLink: **CIBMORCF at RALVM13**
  - Internet: **USIB2HPD@VNET.IBM.COM**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

OS/390 TCP/IP OpenEdition
Configuration Guide

Publication No. SC31-8304-00

If your concern is service related, you can reach Service at 1-800-992-4777 in the United States. Outside the United States, please check your phone listing for the IBM Service Center nearest you.

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	☐	☐

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	☐	☐
Complete	☐	☐
Easy to find	☐	☐
Easy to understand	☐	☐
Well organized	☐	☐
Applicable to your task	☐	☐

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Your Internet Address: _____

Name

Address

Company or Organization

Phone No.

Help us help you!
SC31-8304-00

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Information Development
Department CGMD
International Business Machines Corporation
PO BOX 12195
RESEARCH TRIANGLE PARK NC 27709-9990

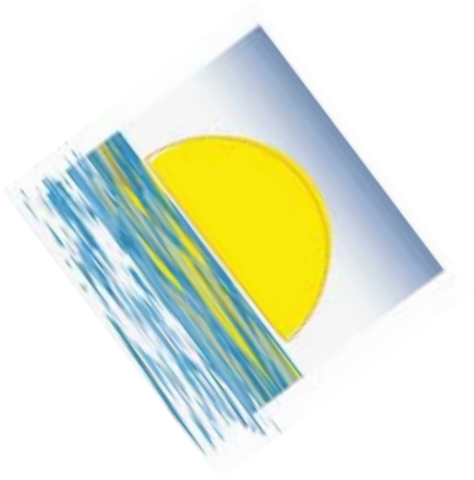
Fold and Tape **Please do not staple** Fold and Tape

SC31-8304-00

Cut or Fold
Along Line

File Number: S390-50
Program Number: 5645-001

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



SC31-8304-00

Spine information:

OS/390 TCP/IP OpenEdition Configuration Guide