

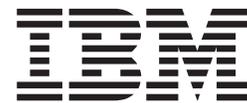
OS/390 eNetwork Communications Server



High Speed Access Services User's Guide

Version 2 Release 6

OS/390 eNetwork Communications Server



High Speed Access Services User's Guide

Version 2 Release 6

Note:

Before using this information and the product it supports, be sure to read the general information under "Appendix. Notices" on page 31.

Second Edition (September 1998)

This edition applies to OS/390 V2R6 (Program Number 5647-A01).

Publications are not stocked at the address given below. If you want more IBM publications, ask your IBM representative or write to the IBM branch office serving your locality.

A form for your comments is provided at the back of this document. If the form has been removed, you may address comments to:

IBM Corporation
Department CGMD
P.O. Box 12195
Research Triangle Park, North Carolina 27709
U.S.A.

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 RALVM17

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.

© Copyright International Business Machines Corporation 1998. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Tables	v
About This Book	vii
Who Should Use This Book	vii
Where to Find Related Information on the Internet	vii
How to Contact IBM Service	viii
High Speed Access Services (HSAS)	1
Functional Characteristics	1
External Interfaces	1
Limitations	4
Setup Requirements	5
Configuration Overview	5
Defining the Stack in the OE Parmlib Member	6
Configuration/Administration Commands	7
oeifconfig—Configuring and Activating a Network Interface for HSAS	8
oeroute—Updating the Network Routing Tables	14
oeping—Testing HSAS Connectivity	18
oenetstat—Displaying Information about HSAS	22
oenetopts—Set Network Options	27
Messages and Codes	29
Appendix. Notices	31
Trademarks	33
Bibliography	35
eNetwork Communications Server for OS/390 V2R6 Publications	35
Softcopy Information	35
Marketing Information	35
Planning	35
Installation, Resource Definition, Configuration, and Tuning	36
Operation	36
Customization	37
Writing Application Programs	37
Diagnosis	38
Messages and Codes	39
APPC Application Suite	39
Multiprotocol Transport Networking (MPTN) Architecture Publications	40
OS/390 Publications	40
Index	41
Readers' Comments — We'd Like to Hear from You.	43

Tables

- 1. Summary of High Speed Access Services Commands 7
- 2. Hex Flag Descriptions 12

About This Book

This book describes how to use the High Speed Access Services (HSAS) functionality of eNetwork Communications Server for OS/390 V2R6 (CS for OS/390). HSAS includes High Speed UDP (previously documented as "HPDT for UDP").

CS for OS/390 is an integral part of the OS/390 V2R6 family of products. For an overview and mapping of the documentation available for OS/390 V2R6, see the *OS/390 Information Roadmap*.

Who Should Use This Book

This book is written for end users and system administrators who want to use applications using a High Speed Access Services connection available in CS for OS/390.

Before using this book, you should be familiar with the IBM Multiple Virtual Storage operating system (MVS). In addition, CS for OS/390 V2R6 should already be installed and customized for your network. For information about installing, see the *Program Directory*. For information about customizing, see *OS/390 eNetwork Communications Server: IP Configuration Guide*.

Where to Find Related Information on the Internet

You may find the following information helpful.

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

You can read more about VTAM, TCP/IP, OS/390, and IBM on these Web pages:

Home Page Uniform Resource Locator (URL)

IBM eNetwork Communications Server

<http://www.software.ibm.com/enetwork/commserver/>

VTAM and TCP/IP

<http://www.software.ibm.com/enetwork/commserver/about/csos390.html>

IBM eNetwork Communications Server Support

<http://www.software.ibm.com/enetwork/commserver/support/>

OS/390

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

IBM

<http://www.ibm.com/>

For definitions of the terms and abbreviations used in this book, you can view or download the latest *IBM Networking Softcopy Glossary* at the following URL:

- <http://www.networking.ibm.com/nsg/nsgmain.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.

High Speed Access Services (HSAS)

High Speed Access Services consists of the following components:

- High Speed UDP (previously documented as "HPDT for UDP"), which supports the following applications:
 - DCE
 - NFS
 - SAP

HSAS supports the ORoutedD dynamic routing application.

High Speed Access Services (HSAS) exploits CS for OS/390's High Performance Data Transfer (HPDT) and Multiprotocol Channel with Common Storage Management (also known as MPC+) and is tightly coupled with the OS/390 UNIX System Services kernel to achieve high performance and minimize CPU utilization. HSAS is managed through local UNIX commands and supports RIPv2 routing.

For application developers, HSAS supports the OS/390 UNIX System Services socket API. Exploiting HSAS should be transparent to socket applications written using the strategic OpenEdition socket library.

Functional Characteristics

High Speed Access Services exclusively uses the efficient and strategic multipath channel (MPC) point-to-point interface to communicate with both enterprise system connection (ESCON) network devices, and Open Systems Adapter (OSA) features. This MPC channel protocol has been designed to minimize the MVS overhead required to get data on to, and off of MVS, while at the same time allowing the MPC resources to be shared by multiple communication stacks (including VTAM).

Socket applications must use the strategic OpenEdition socket library to access HSAS. The common INET feature provided by the OpenEdition socket library allows multiple AF_INET transport drivers to provide access to the IP network as if they were a single entity. From a socket application perspective, the existence of multiple transport drivers is entirely transparent (common INET acts as a pre-router, to direct IP traffic to the transport driver that is 'closest to' the target IP address). However, the HSAS transport driver must be configured, and administrated separately from the full-function stack.

HSAS supports OSA-2 FDDI, OSA-2 Fast Ethernet, or enterprise system connections (ESCON). For OSA-2 connections, the attached node can be any device supporting IP communications over a FDDI or Fast Ethernet LAN. For ESCON connections, the attached node can be either RS/6000, 2216 or OS/390.

External Interfaces

Supported TCP/IP Socket APIs

The following XPG4.2 compliant socket APIs are supported. Refer to the XPG4.2 branding documentation and the OE syscalls manual for the interface details.

- accept() — accept new socket connection
- bind() — assign socket name

High Speed Access Services

- close() — remove socket access for this process
- connect() — restrict communications to specific partner
- gethostid() — obtain the host IP address
- gethostname() — obtain the internal host name
- getpeername() — obtain partner IP address and port
- getsockname() — obtain bound to IP address and port
- getsockopt() — get options set by setsockopt()
- ioctl() — the following ioctls are supported
 - FIONBIO — set/reset nonblock I/O
 - FIONREAD — get number of readable bytes available
 - SIOCATMARK — get Out of Band synchronization mark
 - SIOCGRTTABLE — get network routing table
 - SIOCGIFMETRIC — get network routing metric
 - SIOCGIFNETMASK — get network interface network mask
 - SIOCGIFCONF — get network interface configuration
 - SIOCGIFDSTADDR — get network interface destination address
 - SIOCGIFFLAGS — get network interface flags
 - SIOCADDRT — add a route
 - SIOCDELRT — delete a route
 - SIOCSETRTTD — set socket to be attached to 1 TD
 - SIOCSIFADDR — set network interface address
 - SIOCSIFDSTADDR — set network interface destination address
 - SIOCSIFFLAGS — set network interface flags
 - SIOCSIFMETRIC — set network interface metric
- listen() — specify a queue length for a server
- poll() — multiplex input/output
- read() — read data into a contiguous area (non-fastpath only)
- readv() — read data into a non-contiguous area (non-fastpath only)
- recv() — read data from connected partner (both normal and fastpath)
- recvfrom() — read data into a contiguous area (both normal and fastpath)
- recvmsg() — read data into a non-contiguous area (both normal and fastpath)
- select() — multiplex input/output
- send() — send data to connect partner (both normal and fastpath)
- sendmsg() — send data from a non-contiguous area (both normal and fastpath)
- sendto() — send data from a contiguous area (both normal and fastpath)
- setsockopt() — the following socket options are supported:
 - SO_DEBUG — activate socket level debugging
 - SO_KEEPALIVE — probe idle connections
 - SO_LINGER — socket lingers on CLOSE
 - SO_RCVBUF — set receive buffer size
 - SO_REUSEADDR — allow multiple server to bind to same port concurrently
 - SO_SNDBUF — set send buffer size
- shutdown() — shutdown unidirectional socket access
- socket() — create socket end-point
- write() — send data from a contiguous area (non-fastpath only)

- `writev()` — send data from a non-contiguous area (non-fastpath only)

`Socketpair()` is implemented at the LFS layer.

Optional Fast Path Support

For applications that have **extremely** strict communications path length requirements (i.e., like SAP), an optional extension has been provided to further reduce the HSAS communications overhead. Exploitation of this extension is entirely **optional**. You will obtain very competitive TCP/IP performance by exploiting the base HSAS, without any further understanding of this section.

If you desire to activate this optional extension, an OpenEdition environment variable must be set. This can be done by exporting it in the OpenEdition shell environment before the socket application is invoked, or by the application itself, by invoking the `setenv()` service. The value of the `_BPXK_INET_FASTPATH` environment variable at the time the socket is created, determines whether or not the socket will be marked fast path. To activate the HSAS extension, this environment variable must be set to `OESTACK`.

Note: OpenEdition environment variables have only a process-wide scope (i.e., only affect a single MVS 'job'), so there is no problem if some of your applications exploit fast path services, while others do not.

Once a socket is marked fast path, the communications overhead is reduced on the following socket syscalls:

- `send()`
- `recv()`
- `sendto()`
- `recvfrom()`
- `sendmsg()`
- `recvmsg()`

While extremely efficient, applications are not XPG compliant while using this environment variable (POSIX signals are not supported). Exploiting applications are not interruptible using any signal other than the `SIGKILL` terminating signal and they can not be debugged using the interactive OpenEdition **dbx** debugger. However, an application can be developed and tested using the **dbx** debugger with the environment variable not set and subsequently executed in production with the environment variable set.

You should weigh application needs in terms of performance versus compliance. If XPG compliance is required, the application can still exploit the performance benefits of HSAS without using this environment variable. For a description of how OpenEdition conforms to the XPG portability standard, see the *OS/390 OpenEdition XPG4 Conformance Document*.

This environment variable causes different actions to be taken within the common INET layer depending on the type of socket being created. For UDP, the socket is explicitly bound to the transport driver named in the environment variable (i.e., `OESTACK`). This means that a fast path UDP socket will only be able to communicate with partners that are accessible through `OESTACK` managed network interfaces. So, for UDP, this environment variable effectively overrides the support provided by common INET (i.e., single image TCP/IP). This must be taken into account before activating fast path for a UDP-based application.

High Speed Access Services

No application modifications are required to exploit the fast path environment variable. For more information about setting environment variables, see *OS/390 UNIX System Services Planning*.

Optional Fast Path Support — Restrictions: The following restrictions apply to the fast path environment:

- `onslookup` with the `novc` option does not work if the fast path environment variable is set to `OESTACK`
- fast path sockets do not support:
 - the ability to peek at UDP datagrams on any of the `recv()` syscalls
 - the ability to have a socket implicitly bound as part of a `UDP sendto()`, `sendmsg()`
 - the ability to do bulkmode UDP

Limitations

The following limitations apply to applications using HSAS:

- There is no support for TCP APIs. Clients that access OS/390-resident applications through HSAS must use a different IP address - one that is managed by the full function TCP/IP stack - to access TCP-based applications. However, a single UDP server can satisfy requests from both HSAS and the full function TCP/IP stack concurrently.
- TCP-based applications (such as FTP, TELNET, RLOGIN, SMTP, WWW, and DRDA*) cannot use HSAS.
- The OpenEdition common INET physical file system (PFS) is required to support communications with both TCP-based and UDP-based clients.
- Applications can use the new `_BPXK_INET_FASTPATH` environment variable to further reduce CPU overhead. However, this support is not XPG 4.2 compliant (POSIX*** signals are not supported). For more information, see “Supported TCP/IP Socket APIs” on page 1.
- No router or firewall support is provided by the HSAS stack.
- General purpose `IP_RAW` support is not provided. However, the common network problem determination tools (such as *ping* and *netstat*) are provided (as *oeping* and *oenetstat*).
- Multicast IP addresses are not supported.
- The HSAS stack supports a maximum of 32 incoming fragments. Consequently, this stack cannot properly handle any incoming datagrams which must be broken up into more than 32 fragments (based on the MTU size defined for the interface). Therefore, be careful about defining an MTU size that is too small. The *oeping* command can be used to verify that a given size of data can be handled successfully over a given interface.
- The HSAS stack cannot be stopped and restarted. The only way to restart the HSAS stack is via a re-IPL.
- SRB mode or asynchronous callers are not supported for High Speed UDP.

Also, HSAS does not support:

- Virtual IP Addressing (VIPA)
- Simple Network Management Protocol (SNMP) subagent
- packet trace
- supernetting/Classless Inter-Domain Routing (CIDR)

- WLM registration
- AnyNet SNA over IP
- samehost Point-To-Point (PTP) connections (IUTSAMEH)

Setup Requirements

If you are using the OSA-2 feature, you must make sure you have the minimum code level installed. For an OSA-2 FDDI feature, the level is 6.10 and for an OSA-2 Fast Ethernet feature, the base GA level will support the HPDT MPC mode. You also need the OSA Support Facility Version 1 Release 2 with APAR 0W28254/PTF UW40550 to configure the OSA feature to run in the HPDT MPC mode.

Over an ESCON channel, this support uses the HSAS MPC channel protocol (MPC+). If you are using ESCON to connect to a remote AIX, you need one of the following versions of the ESCON device driver for AIX:

- V1R1.0.5 with PTF associated with APAR U447758
- V1R1.0.0 with PTF associated with APAR U449691

The remote AIX must be running V4R3 or higher.

Because VTAM controls the channel-level resources, you must define a VTAM transport resource list entry (TRLE) to exploit the HSAS communications stack. For more information, see the *OS/390 eNetwork Communications Server: SNA Resource Definition Reference* or you can obtain the TRLE name used for the HSAS network interface from your VTAM administrator.

To use HSAS when running OS/390 as a guest operating system, you need the following VM PTFs:

- UM28199
- UM28200
- UM28201

Configuration Overview

Use the following steps to set up High Speed Access Services:

1. Install appropriate software as listed in “Setup Requirements”.
2. If you are using HSAS over an ESCON channel, define the OS/390 I/O configuration for the ESCON channel as described in the *OS/390 HCD User's Guide*. The ESCON-attached node must also be configured.
3. Add a new SUBFILESYSTYPE definition statement for HSAS to the OpenEdition parmlib member, BPXPRMxx. For more information, see “Defining the Stack in the OE Parmlib Member” on page 6.
4. Make sure you have a VTAM TRLE defined and activated for the ESCON channel or OSA-2 connection. The MAXBFRU operand must be set to 9 in order for that TRLE to support HSAS. The TRLE name is used to configure the HSAS network interface on the *oeifconfig* command. You activate the TRLE using the *VARY ACT* command. Note that if the TRLE is not enabled for HPDT support (MPCLEVEL=NOHPDT operand is coded), the connection cannot support HSAS.

High Speed Access Services

Information about defining the TRLE can be found in the *OS/390 eNetwork Communications Server: SNA Resource Definition Reference*. For more information about VTAM commands, see *OS/390 eNetwork Communications Server: SNA Operation*.

5. Configure and activate the HSAS network interface using the *oeifconfig* command. For more information, see “oeifconfig—Configuring and Activating a Network Interface for HSAS” on page 8.

Defining the Stack in the OE Parmlib Member

You must define the OpenEdition Common INET physical file system to include the HSAS communications stack (OESTACK). Whereas other IP communications stacks require a separate address space, the HSAS communications stack resides within the OpenEdition kernel. Define the HSAS communications stack through the BPXPRMxx member in SYS1.PARMLIB. The following is a sample for BPXPRMxx:

```
FILESYSTYPE    ENTRYPOINT(BPXCINT)
                TYPE(CINET)
SUBFILESYSTYPE NAME(TCPIP)
                ENTRYPOINT(EZBPFINI) TYPE(CINET)
                DEFAULT
SUBFILESYSTYPE NAME(OESTACK)
                ENTRYPOINT(BPXUIINT)
                TYPE(CINET)
NETWORK DOMAINNAME(AF_INET)
                DOMAINNUMBER(2)
                MAXSOCKETS(25000)
                TYPE(CINET)
                INADDRANYPORT(20000)
                INADDRANYCOUNT(4000)
```

Where:

FILESYSTYPE

defines the common INET physical file system (PFS), which is required to allow multiple AF_INET transport drivers to provide IP network access transparently from a socket application perspective.

SUBFILESYSTYPE NAME(TCPIP) statement

defines the full-function TCP/IP stack. Its parameter, **DEFAULT**, makes TCP/IP the default stack, since it is the only full-function stack. The **NAME** parameter can be variable for TCP/IP. Refer to *OS/390 UNIX System Services Planning* for more information.

SUBFILESYSTYPE NAME(OESTACK) statement

defines the HSAS stack. You must code this statement exactly as shown (including the identifier OESTACK) and you may code only one such statement.

NETWORK statement

correlates the common INET PFS with all its sub-transport drivers through the TYPE parameter. The INADDRANYPORT and INADDRANYCOUNT parameters reserve ports explicitly for common INET use. Refer to *OS/390 UNIX System Services Planning* for more information about INADDRANYPORT and INADDRANYCOUNT.

Configuration/Administration Commands

This section describes commands that are used to configure and display information about High Speed Access Services. All commands are issued from the OpenEdition/MVS shell. Issuing a command with no parameters displays the syntax of the command (except *oenetstat*, for which a "?" as a parameter will retrieve the syntax).

Because settings by the *oeifconfig* and *oeroute* configuration commands are for the duration of an IPL, the best place for them is in the */etc/rc* shell script, so the OpenEdition initialization always configures the HSAS stack. For more information about customizing the OpenEdition shell, as well as the sample */etc/rc* script, see *OS/390 UNIX System Services Planning*.

Table 1. Summary of High Speed Access Services Commands

Task	Command
Configure the network interface for High Speed Access Services	<i>oeifconfig</i>
Activate or deactivate a network interface for High Speed Access Services	<i>oeifconfig</i>
Add or delete entries in the network routing table	<i>oeroute</i>
Test High Speed Access Services connectivity to a host	<i>oeping</i>
Display information about a network interface	<i>oeifconfig</i>
Display information about the local host	<i>oenetstat</i>
Tune the network by changing network options	<i>oenetopts</i>

oeifconfig—Configuring and Activating a Network Interface for HSAS

Purpose

The *oeifconfig* command assigns an IP address to the network interface and configures network interface parameters. A network interface must be configured and activated before applications can use HSAS. This command can be used to perform the following specific tasks:

- Configure an OSA FDDI network interface for HSAS
- Configure an OSA Fast Ethernet network interface for HSAS
- Configure an ESCON network interface for HSAS
- Activate, deactivate, or remove the configuration of an HSAS network interface
- Query the status of an HSAS network interface

If the **down** parameter is not specified on the initial configuration, the network interface is initially activated. If you want to make changes to a network interface, you must first remove the existing configuration by issuing the *oeifconfig* command with the **detach** parameter.

When a network interface is configured on the *oeifconfig* command, the command also establishes a route to local host(s) based on the value of the netmask parameter. These routes can be displayed using *oenetstat* and can be removed only by using *oeifconfig detach*. Routes created by the *oeifconfig* command cannot be deleted by an *oeroute delete* command.

oeifconfig processing operates asynchronously. A subsequent *oenetstat -i* or *oeifconfig* command (using the query format) can be used to determine the current attributes of the interface and whether it is up or not. Also, any link communication errors during previous *oeifconfig* processing are reported on the MVS console and logged as VTAM or OpenEdition messages.

Format

Configure an OSA FDDI or OSA Fast Ethernet network interface for HSAS:

►► *oeifconfig*— *interface_name* —

inet
inet

 — *source_IP_address* —————►

►- *osaport*— *OSA_port_number*— *mtu*— *maximum_transmission_unit* —————►

►| *inet optional parameters* |—————►

inet optional parameters

| —

checksum
checksum
nochecksum
-checksum

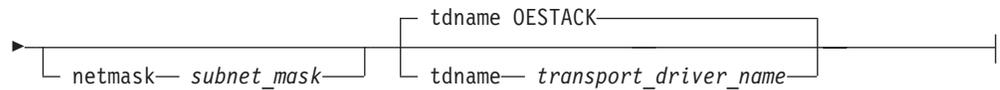
 —

up
down

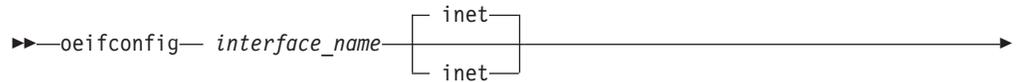
 —

metric 0
metric <i>number</i>

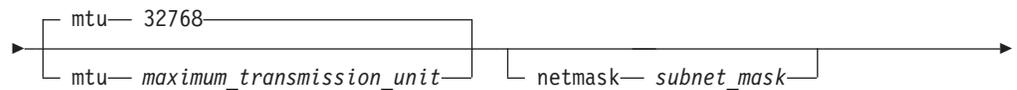
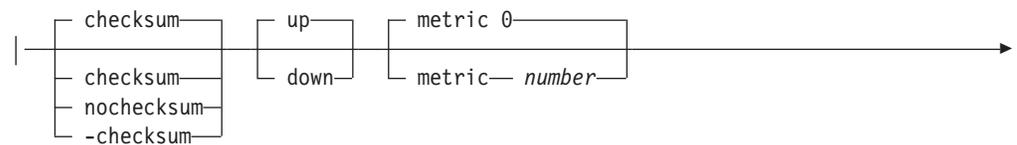
 —————►



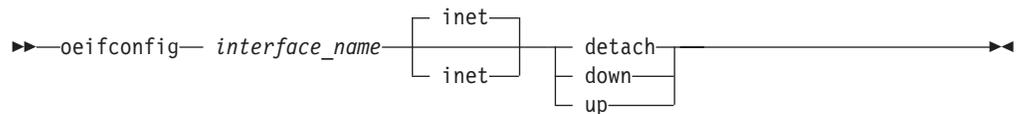
Configure an ESCON network interface for HSAS:



inet optional parameters



Activate, deactivate, or remove an existing HSAS network interface:



Query the status of a HSAS network interface:



Usage

Since *oeifconfig* processing is asynchronous, it is necessary to test the results of an *oeifconfig* configure command by subsequently issuing an *oeifconfig* query command and checking the exit value. For example,

```
# try oeifconfig every 60 seconds until interface is up
oeifconfig tr1e1 121.1.2.3 121.1.2.9
until oeifconfig tr1e1
do
    sleep 60
    oeifconfig tr1e1 up
done
```

This command might typically be issued from the `/etc/rc` script that executes when OpenEdition initializes. Superuser authority is required to configure, activate, deactivate, or remove the configuration of a HSAS network interface.

High Speed Access Services

Note: VTAM must be active in order to activate a network interface with HSAS.

Parameters

address_family

specifies the *address_family* for this interface. The only supported *address_family* for the *oeifconfig* command is **inet**, which is the default.

checksum

-checksum

nochecksum

determines whether checksum support is enabled for this network interface. Checksum support indicates whether or not a reliable communications path exists between the communicating applications. Specify **checksum** if any portion of the path between the communicating applications is unreliable (such as a LAN or WAN). Specify **-checksum** (or **nochecksum**) if the entire path is reliable (such as ESCON or an SP2 fast switch). The **nochecksum** and **-checksum** are functionally equivalent.

This parameter is valid only when configuring the network interface. The default is **checksum**.

destination_IP_address

specifies the IP address of the destination of a point-to-point connection (for example, an ESCON-channel attached RISC/6000). The IP address can be expressed either in dotted decimal format (for example, 9.117.81.65) or as a symbolic name known to a domain name server (for example, abc.def.ibm.com).

detach

deactivates and removes a previously configured interface. After the **detach** parameter is used, the network interface cannot be activated until it is reconfigured by another *oeifconfig* command.

down

deactivates a network interface. The interface can be reactivated by issuing another *oeifconfig* command and specifying the **up** parameter. A network interface can be configured using the **down** parameter and then subsequently activated with the **up** parameter.

inet

specifies the *address_family* for this interface. The only supported *address_family* for the *oeifconfig* command is **inet**, which is the default.

interface_name

specifies the 1- to 8- character name of the interface corresponding to the label specified on the VTAM TRLE definition statement for the connection.

metric number

specifies the metric number (number of hops) for the interface in a range from 0 to 16. The OpenEdition common INET component uses this number to determine the best route to a host. This parameter is valid only when configuring the network interface.

mtu maximum_transmission_unit

specifies the maximum transmission unit size for the interface. All hosts connected to the same ESCON link or OSA LAN must agree on a common maximum transmission unit size. This parameter is valid only when configuring the network interface.

High Speed Access Services

For the OSA FDDI network interface, the range of this value is 512 to 4050 bytes. For OSA Fast Ethernet, the range is 512 to 1492. This parameter is required when configuring the OSA interface.

For ESCON-attached devices, the range of this value is from 512 to 32768 bytes. The default is 32768 bytes.

netmask *subnet_mask*

specifies the subnet mask associated with the interface. The value can be expressed in either dotted decimal format (for example, 255.255.255.0), or in hexadecimal format (for example, 0xfffff00). This parameter is valid only when configuring the network interface. If netmask is not specified, the subnet mask is based on the local address portion of the destination address.

For a complete description of how subnetting works, see *TCP/IP Tutorial and Technical Overview*.

osaport *port_number*

specifies the OSA port number of this interface and identifies the external attachment to a LAN supported by the OSA using the IP protocol. Valid port number assignments for OSA range from 0 to 16.

Note: 0 is the only port number supported by HSAS.

source_IP_address

specifies the OS/390 home IP address of this interface. This address must be unique for the entire OS/390 image, which includes the full-function TCP/IP stack. The IP address can be expressed either in dotted decimal format (for example, 9.117.81.65) or as a symbolic name known to a domain name server (for example, abc.def.ibm.com).

tdname *transport_driver_name*

specifies the name of the transport driver used by the network interface. This value corresponds to the name on the SUBFILESYSTYPE statement of the BPXPRMxx parmlib member. The only supported value is **OESTACK**.

up activates a network interface. The **up** parameter is valid only when the interface is configured and not already active.

Results

Note: The results below are for the non-query version of the command.

The configure (i.e., non-query) version of the command normally results in no response to the shell environment, unless an immediate error is detected. However, an operator message will be written once the asynchronous configuration process is complete. An operator message is generated for both successful, and unsuccessful configuration attempts. You can use *oenetstat -i*, or *oeifconfig* (query version) to determine the current interface status.

Note: The results below are for the query version of the command.

If the *oeifconfig* command is issued with only an *interface_name*, the resulting display shows:

- Attribute flag bytes in hex. The flag bytes are described in the net/if.h header file.
- The IP address of the interface
- Netmask associated with the interface

High Speed Access Services

- IP address of the destination or the OSA port number associated with this interface
- The link initialization error message identifier and its corresponding reason code, if an error occurred initializing the interface
- Names of the attributes that are enabled with their associated hex flag bytes:

Table 2. Hex Flag Descriptions

Hex flag	Attribute	Description
X'001'	UP	Indicates that the network interface is active.
X'002'	BROADCAST	Indicates that the broadcast address is valid.
X'004'	DEBUG	Indicates that debugging is on.
X'008'	LOOPBACK	Indicates that the interface is a loopback net.
X'010'	POINTTOPOINT	Indicates that the interface is point to point.
X'040'	RUNNING	Indicates that resources are allocated.
X'080'	NOARP	Indicates that the interface does not use address resolution protocol support to map IP addresses to hardware addresses.
X'400'	CHECKSUM	Indicates that the checksum support is enabled. Checksum support indicates whether or not a reliable communications path exists between the communicating applications.

If an error occurred initializing the interface, this command displays the link initialization error message identifier and the corresponding reason code. The error is specified as either a VTAM error or an OpenEdition error. VTAM errors are documented in *OS/390 eNetwork Communications Server: SNA Messages*. OE errors and message identifiers are documented in *OS/390 UNIX System Services Messages and Codes*.

Examples

1. Configuring a High Speed Access Services ESCON Network Interface

The following command configures and activates an ESCON network interface.

```
oeifconfig trle1 9.115.21.21 9.115.21.20
```

In this example, the network interface **trle1** supports a home IP address of 9.115.21.21. The interface provides a point to point connection with 9.115.21.20.

2. Configuring a High Speed Access Services OSA Network Interface

The following command configures and activates an OSA network interface.

```
oeifconfig trle2 9.116.21.21 osaport 0 mtu 1500 netmask 0xfffff00
```

In this example, the network interface **trle2** supports a home IP address of 9.116.21.21 using port 0 on an OSA adapter. This particular adapter is identified in the VTAM configuration under the name TRLE2. Local IP destinations using the form 9.116.21.* can be reached through this interface.

3. Querying the HSAS Network Interface

High Speed Access Services

The following command displays the status and attributes of the network interface, **trle5**.

```
oeifconfig trle5
```

The command returns the following output:

```
trle5: flags= d1 < UP POINTOPOINT RUNNING NOARP >
      inet 9.38.49.219 netmask 0xff000000 destination 9.38.49.220
```

If an error occurred when the interface was initialized, the command would return output as follows:

```
trle5: flags= d0 < POINTOPOINT RUNNING NOARP >
      inet 9.38.49.214 netmask 0xff000000 destination 9.38.49.215
      link-error message=BPXU003I (OE code = 0x1071042c)
```

In this example, the link-error message identifies an OpenEdition message number and reason code issued in response to the error.

4. **Activating a HSAS Network Interface**

The following command activates a previously configured network interface named **trle1**.

```
oeifconfig trle1 up
```

5. **Deactivating a HSAS Network Interface**

The following command deactivates but does not unconfigure a network interface named **trle1**.

```
oeifconfig trle1 down
```

The following command deactivates and unconfigures an interface named **trle1**.

```
oeifconfig trle1 detach
```

Return Codes

Here are the exit values for the *oeifconfig* command:

Exit Value	Description
0	configure version of the command successfully initiated
0	query version of the command indicates configuration completed successfully
1	command failed — an error message is displayed
2	query version of the command indicates the interface is not up

High Speed Access Services

oeroute—Updating the Network Routing Tables

Purpose

The *oeroute* command is used to add or delete an entry in the High Speed Access Services routing table. Routes are typically added to connect to remote hosts. Since *oeifconfig* builds the routes for the local hosts, *oeroute* is only necessary for remote connections beyond the ESCON link or OSA LAN.

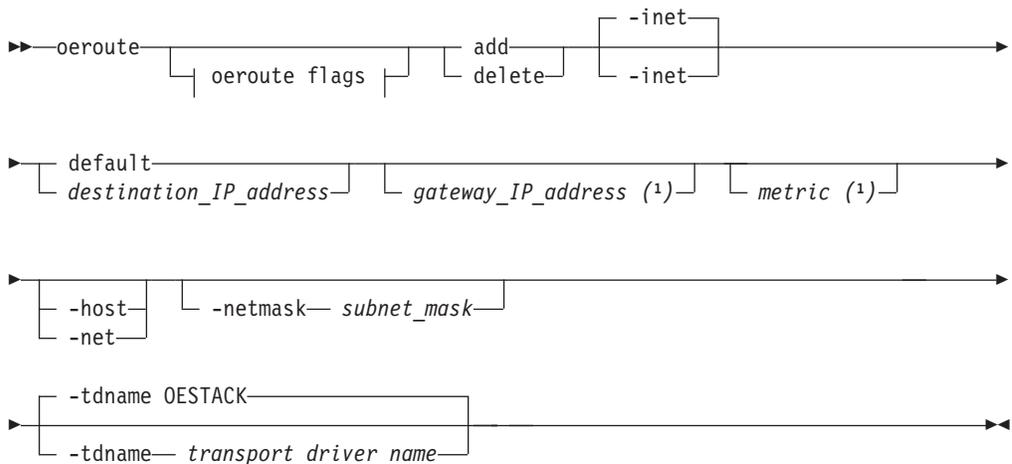
Routes cannot be added to a configuration unless HSAS can resolve that route to a known gateway. In general, this means that an *oeroute* command must have been preceded by a corresponding *oeifconfig* command which defines an interface that will be part of that route.

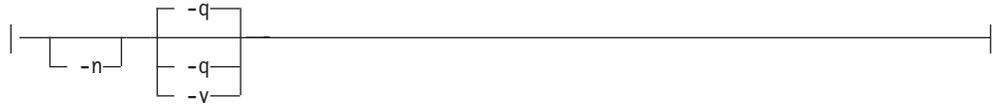
The *oeroute* command distinguishes between routes to hosts and routes to networks by interpreting the destination IP address, which can be specified either by symbolic name or numeric address. The *oeroute* command attempts to resolve all symbolic names into addresses. If the destination has a host address port of 0 or if the destination is the symbolic name of a network, then the route is assumed to be to a network. Otherwise, the destination is presumed to be a route to a host. You can force the destination to be interpreted as a network or as a host by specifying the optional keywords **-net** and **-host**.

If you need to determine the current contents of the routing table, issue *oenetstat -r*. See “*oenetstat—Displaying Information about HSAS*” on page 22 for more information about this command.

When a network interface is configured on the *oeifconfig* command, the command also establishes a route to local host(s) based on the value of the netmask parameter. These routes can be removed only by using *oeifconfig detach*. Routes created by the *oeifconfig* command cannot be deleted by an *oeroute delete* command.

Format



oeroute flags**Notes:**

1. Valid only when adding a route.

Flags

- n** indicates that numeric IP addresses rather than symbolic names are displayed in any output from the *oeroute* command.
- q** suppresses output of the *oeroute* command (quiet mode).
- v** displays information as output for the *oeroute* command (verbose mode).

Usage

This command might typically be issued from the `/etc/rc` script that executes when OpenEdition initializes. Superuser authority is required to update the HSAS network routing table.

Parameters**add**

adds a new route to the HSAS routing table.

-address family

indicates the address family for this route. **-inet** is the only supported address family and is the default.

delete

deletes an existing route from the HSAS routing table.

destination_IP_address

specifies the IP address of the destination for the route. The address can indicate either a destination host or a destination network. The IP address can be expressed either in dotted decimal format (for example, 9.117.81.65) or as a symbolic name known to a domain name server (for example, abc.def.ibm.com). A default route can be indicated by specifying **default** instead of an IP address.

gateway_IP_address

specifies the IP address of the gateway or next hop to reach the destination address. The IP address can be expressed either in dotted decimal format (for example, 9.117.81.65) or as a symbolic name known to a domain name server (for example, abc.def.ibm.com). This value can only be specified when adding a route.

-host

forces the destination address to be treated as a host address rather than a network address. This parameter is incompatible with the **-net** and **-netmask** parameters.

metric

specifies the metric number (number of hops) associated with this route (from 0 to 16). The OpenEdition common INET component uses this number to determine the best route to a host. The default value is 0. If a metric is

High Speed Access Services

specified, it must be placed after the gateway and destination IP addresses. This value can only be specified when adding a route.

-net

forces the destination address to be treated as a network address rather than a host address.

-netmask *subnet_mask*

specifies the subnet mask associated with the destination address. The value can be expressed in either dotted decimal format (for example, 255.255.255.0), or in hexadecimal format (for example, 0xfffff00). If no subnet mask is specified, the subnet mask is based on the host address portion of the destination address, as well as the setting of the **-net** or **-host** parameter. Depending on the particular subnet mask used, it may be necessary to supply the exact same netmask for *oeroute delete* that was used for the corresponding *oeroute add*.

Do not confuse this parameter with subnet mask assigned to a local interface using the *oeifconfig* command. **-netmask** here applies only to the destination address(es) of the route being defined.

For a complete description of how subnetting works, see *TCP/IP Tutorial and Technical Overview*.

-tdname *transport_name*

specifies the name of the transport driver (corresponds to the name on the SUBFILESYSTYPE statement of the BPXPRMxx parmlib member) that processes the *oeroute* command. The only supported value is **OESTACK**.

Examples

1. Adding a HSAS route with a gateway

The following command adds a route to 212.1.2.36 using 8.39.49.215 as the gateway with a metric count of 4.

```
oeroute -v add 212.1.2.36 8.39.49.215 4
```

The command returns the following output:

```
Destination address = 212.1.2.36 - Gateway address = 0AIXMPCJ
```

2. Adding a default gateway

The following command adds the default gateway, 9.117.80.252.

```
oeroute add default 9.117.80.252
```

3. Adding a network route specifying a range of destination IP addresses

The following command adds a network route (that is, all IP addresses that are in the range 9.117.80.1 to 9.117.81.255) through gateway 9.117.85.65 .

```
oeroute add 9.117.80.0 9.117.85.65 1 -netmask 0xfffffe00
```

4. Deleting a network route

The following command deletes the network route added by the command in the previous example.

```
oeroute delete 9.117.80.0 -netmask 0xfffffe00
```

Results

For *oeroute* quiet mode, there is no display. For *oeroute* verbose mode, the destination and gateway addresses are displayed.

Return Codes

Here are the exit values for the *oeroute* command:

Exit Value	Description
0	command completed successfully
1	command failed — an error message is displayed

oe ping—Testing HSAS Connectivity

Purpose

The *oe ping* command sends an echo request to obtain an echo response from a host or gateway. The *oe ping* command is useful for:

- Determining the status of the network and various foreign hosts
- Tracking and isolating hardware and software problems
- Testing, measuring, and managing networks

If the host is operational and located in the network, it responds to the echo. Each echo request contains an Internet Control Message Protocol (ICMP) header, and enough bytes to fill out the packet.

The *oe ping* command will not support pinging any of the IP addresses that are local to the MVS image. It can only be used to access remote IP addresses.

The *oe ping* command sends one datagram per second and prints one line of output for every response received. The *oe ping* command calculates round-trip times and packet loss statistics, and displays a brief summary on completion. This command completes when either the program times out or a SIGINT signal is received. The *host* parameter is either a valid host name or internet address.

By default, the *oe ping* command continues to send echo requests to the display until a signal is received (for example, ASCII Ctrl-C). The Interrupt key can be changed by using the *stty* command. See the *OS/390 UNIX System Services Command Reference* for more information.

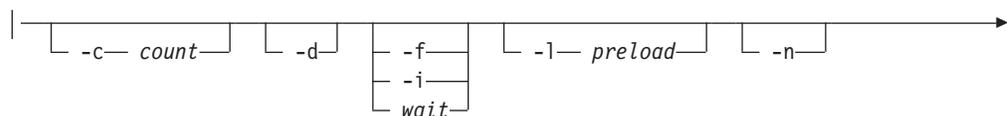
Because of the load that continuous echo requests can place on the system, repeated requests should be used primarily for problem isolation.

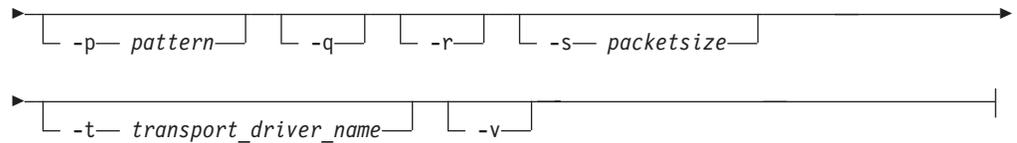
Note: The *oe ping* command will attempt to use the specific name specified with the **-t** flag as the transport driver. This means *oe ping* will fail when using destination IP addresses with routes not available via that transport driver. HSAS, with the name OESTACK, is the default transport driver if the **-t** flag is not specified. Other transport driver names can be selected using this flag. (A warning message is issued if the transport driver is not available, and then the default stack defined to Common Inet is chosen.) The default behavior of *oe ping* is in contrast to the *oping* command, which by default will go to the transport driver selected by the Common Inet pre-router.

Format

►► *oe ping* | *oe ping* flags | *host* ◀◀

oe ping flags





Usage

This command uses the ICMP protocol. If you experience any problems using *oe ping*, examine */etc/protocols* to ensure that the ICMP statement is specified.

Note: This version of *ping* is ported from the copyrighted version at the University of California, Berkeley. The **-R** flag is not supported for this version of *ping*.

Flags

-c *count*

specifies the number of echo requests to be sent and received.

-d starts socket-level debugging. This flag has no effect on HSAS.

-f specifies the flood-ping option. This flag displays packets as fast as they come back or one hundred times per second, whichever is faster. For every ECHO_REQUEST sent, a period is printed and for every ECHO_REPLY received, a backspace is printed. This provides a rapid display of how many packets are being dropped. Only the root user may use this option.

Note that this can affect network performance and should be used with caution. Flood-pinging the broadcast address is not permitted. The **-f** flag is not compatible with the **-i** flag.

-i *wait*

waits the number of seconds between sending each packet. The default is to wait for one second between each packet. This option is not compatible with the **-f** flag.

-l *preload*

sends the number of packets specified by *preload* as fast as possible before applying the normal send rate, which is one packet per second. For meaningful results, the value for this flag should be greater than the value specified for **-c**.

-n specifies numeric output only. No attempt is made to look up symbolic names for host addresses.

-p *pattern*

specifies up to 16 pad bytes to fill out the packet sent on the echo request. This is useful for diagnosing data-dependent problems in a network. For example, **-p ff** fills the packet with all 1s.

-q specifies quiet output. Nothing is displayed except the summary lines at start-up time and finish time.

-r bypasses the routing tables and sends directly to a host on an attached network. If the host is not on a directly connected network, the *oe ping* command generates an error message. This option can be used to ping a local host through an interface that no longer has a route through it.

-s *packetsize*

specifies the number of data bytes to be sent. The default is 56, which translates into 64 ICMP data bytes when combined with the 8 bytes of ICMP header data.

High Speed Access Services

- t *transport_driver_name*
specifies the name of the transport driver used for the *oeeping* command. This value corresponds to the name on the SUBFILESYSTYPE statement of the BPXPRMxx parmlib member. **OESTACK** (which indicates HSAS) is the default transport driver for *oeeping*.
- v requests verbose output, which lists ICMP packets that are received in addition to echo responses.

Parameters

host

specifies the host to be pinged (where the echo request is to be sent). This can be specified either as an IP address in dotted decimal format (for example, 9.117.81.65) or as a symbolic name known to a domain name server (for example, abc.def.ibm.com).

Results

The resulting display shows:

- The IP address of the target host
- The number of bytes in the packet, including the ICMP header.
- A packet sequence number
- The maximum number of IP routers that the packet can be transmitted before being discarded (ttl)
- The calculated round trip time

Other information can be displayed depending on the flags specified on the command.

Examples

1. Specifying the Number of Echo Requests

The following command sends two *oeeping* requests to host 9.38.49.217 and waits 5 seconds between each transmission.

```
oeeping -c 2 -i 5 9.38.49.217
```

The command returns the following output:

```
5 , , 5
OEPING 9.38.49.217 (9.38.49.217): 56 data bytes
64 bytes from 9.38.49.217: icmp_seq=0 ttl=51 time=6.705 ms
64 bytes from 9.38.49.217: icmp_seq=1 ttl=51 time=6.829 ms
```

```
--- 9.38.49.217 oeeping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 6.705/6.766/6.829 ms
```

2. Bypassing the Routing Table

The following command sends the request directly to host 9.38.49.217, bypassing the routing table. The **-p** flag fills the ICMP packet with the hexadecimal pattern aa.

```
oeeping -c1 -paa -rq 9.38.49.217
```

The command returns the following output:

```
PATTERN: 0xaa
OEPING 9.38.49.217 (9.38.49.217): 56 data bytes
```

3. Using the Preload Option

High Speed Access Services

The following command uses the `preload` option and shows how the flags can be combined to obtain peculiar output.

```
oeeping -15 -c1 9.38.49.217
```

The command returns the following output.

```
OEPING 9.38.49.217 (9.38.49.217): 56 data bytes
64 bytes from 9.38.49.217: icmp_seq=0 ttl=51 time=13.705 ms

--- 9.38.49.217 oeeping statistics ---
6 packets transmitted, 1 packets received, 83% packet loss
round-trip min/avg/max = 13.705/13.704/13.705 ms
```

Notice the amount of packet loss reported in the output for the previous command. This is not due to a network problem — instead it is caused by conflicting flag values. For best results, always set the *count* value greater than the *preload* value when the `-c` and `-l` flags are used in the same command.

Return Codes

Here are the exit values for the `oeeping` command:

Exit Value	Description
0	command completed successfully
1	command failed — an error message is displayed

High Speed Access Services

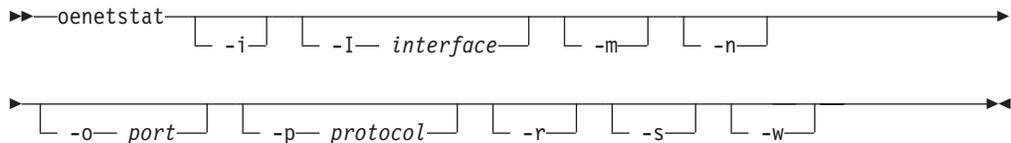
oenetstat—Displaying Information about HSAS

Purpose

The *oenetstat* command displays High Speed Access Services network information gathered from the internal data structures of the HSAS communication stack.

This command can be used for gathering information for diagnostic or tuning activities.

Format



Note: You may specify many options. *oenetstat* will choose one or more options and ignore others when the options conflict. For example, the **-m** option overrides the **-i** option.

Flags

- i** Shows the state of all configured interfaces.
- I interface**
Shows the state or statistics of a specific configured interface.

Note: **-I** can be used by itself, with **-w** and/or with **-s**.
- m** shows memory-related statistics.
- n** Shows network addresses as numbers. When this flag is not specified, the command interprets addresses and displays them symbolically where possible. If this is not possible, then numbers in dotted decimal format are used instead. A Domain Name Server or a local resolution file (such as `/etc/hosts`) must be available for network addresses to be resolved symbolically. If neither facility is available, the use of **-n** is recommended to improve performance.
- r** Shows routing table data.
- s** Shows packet or routing related statistics for different protocols.
- o port**
Specifies the port number to be used to limit the display to a specific local port. This applies to the default display.
- p protocol**
Specifies the name of the protocol to be supplied for statistical display. The **-s** flag is implied when the protocol is supplied.
- w** Shows performance statistics for each interface. These statistics are unique to HSAS and are useful only for diagnostic and internal tuning purposes. They are subject to change and no attempt is made to interpret them in this publication.

Results

The sections show the information displayed by various display queries.

Default Display (oenetstat without flags)

The default display shows the following active sockets data:

- OpenEdition process identifier for the socket.
- Protocol (An asterisk following the protocol indicates that fast path mode is in effect for this connection.)
- Local address
- Remote address, if a connection was established

Internet addresses are of the form *IP_address.port*. The *IP_address.port* will be converted to a symbolic name if the **-n** flag is not specified, and the symbolic name is found. Unspecified addresses and ports are printed as an asterisk (*).

Interface display (oenetstat -i or -l)

The interface display format shows the following data for each interface:

- Interface name
- Maximum transmission unit (*mtu*) size
- Interface IP address
- Number of incoming packets received and those in error.
- Number of outgoing packets sent and those in error.
- Interface state
- Maximum blocking factor attained for this interface
- Current blocking factor for this interface

Note: Blocking factor represents the output load on the interface. A higher value indicates a greater load.

Routing Table Display (oenetstat -r)

The routing table display format indicates the available routes and their statuses. Each route consists of a destination host or network and a gateway to use in forwarding packets.

The routing table display format shows the following routing data:

- Flags indicating the status and type of route.
 - U – route is up
 - G – route to destination is indirect (i.e., gateway is used)
 - H – route is to a host
 - L – route is local to this host
 - S – route is a static route added using the *oeroute* command
- Gateway address of the outgoing interface, or '*' if the route is to a host on the local network.
- Reference count of active uses for the route (always = 0).
- A count of the number of packets sent using that route.
- Interface name corresponding to the route.

High Speed Access Services

- Number of destinations found unreachable.

Statistics Display (`oenetstat -s`)

This format displays protocol usage statistics data. The values are internally stored in fields appropriate with their intended usage – full words or double words. Be aware, however, that wrapping of the values is possible. This should be an uncommon event in most normal situations. Detaching an interface will reset the statistical counts for that interface to 0.

Protocol usage for IP includes:

- Number of total packets received
- Number of packets sent from this host
- Number of with bad header checksums
- Number of with size smaller than minimum
- Number of fragments received
- Number of fragments dropped (timeout) (all interfaces only)
- Number of packets reassembled ok
- Number of packets for an unknown/unsupported protocol
- Number of output packets dropped (no space)
- Number of output packets dropped (no route) (all interfaces only)
- Number of output datagrams fragmented
- Number of broadcast/multicast packets dropped

Protocol usage for ICMP includes:

- Number of output echo replies
- Number of output echo replies dropped (no route)
- Number of output destination unreachable
- Number of bad checksums
- Number of messages with bad length
- Number of input echo replies
- Number of input echos
- Number of input timeout exceeded

Protocol usage for UDP includes:

- Number of datagrams received
- Number of datagrams sent
- Number of bad checksums
- Number of incomplete headers
- Number of bad data length fields
- Number of datagrams dropped (no socket)
- Number of socket buffer overflows
- Number of broadcast/multicast dropped due to no socket
- Number of unexpected enhanced-UDP datagrams received

Memory Usage Display (`oenetstat -m`)

Memory usage data includes:

- High water mark for I/O write buffers
- Current maximum I/O write buffers
- Current I/O write buffers in use
- High water mark for I/O read buffers
- Current maximum I/O read buffers
- Current I/O read buffers in use

Examples

- The following command provides the default display.

```
oenetstat
```

The command returns the following output:

Process-Id	Protocol	Local-Addr	Remote-Addr	Flags
100663302	udp	9.38.49.214.1001	*.*	F
100663302	udp	9.38.49.214.9000	*.*	F
100663302	udp	9.38.49.214.10001	*.*	
100663302	udp	OMVSOE2D.18121	*.*	

- The following command requests memory usage:

```
oenetstat -m
```

The command returns the following output:

```
write I/O buffers:
    current usage =      5
    high water mark =     13
    current maximum =     64
read I/O buffers:
    current usage =      1
    high water mark =      1
    current maximum =     64
```

- The following command requests information for all configured network interfaces. The **-n** flag specifies that the IP address is returned in numerical form.

```
oenetstat -in
```

The command returns the following output:

Name	Mtu	Address/ Network	Ipkts	Ierrs	Opkts MaxBlock	Oerrs Block
L00	32768	127.0.0.1 127.0.0.0	0	0	0 1	0 up 1
TRLE3	1500	9.38.50.111 9.0.0.0	0	0	0 1	0 1
TRLE4	32768	9.38.49.214 9.0.0.0	13456	17	69455 3	33 up 3

- The following command shows packet and routing related statistics. You can limit the display output to show statistics only for a particular network interface (using the **-I** flag) or for a particular protocol (using the **-p** flag).

```
oenetstat -s
```

The command returns the following output:

```
ip:
    42931 total packets received
    44130 packets sent from this host
        5 with bad header checksums
        0 with size smaller than minimum
    7210 fragments received
    450 fragments dropped (timeout)
```

High Speed Access Services

```
7110 packets reassembled ok
 24 packets for an unknown/unsupported protocol
  0 output packets dropped (no space)
112 output packets dropped (no route)
  0 output datagrams fragmented
  0 broadcast/multicast packets dropped
```

icmp:

```
13 output echo replies
  0 output echo replies dropped (no route)
  0 output destination unreachable
  0 bad checksums
  0 messages with bad length
39 input echo replies
  0 input echos
  0 input timeout exceeded
```

udp:

```
39319 datagrams received
40047 datagrams sent
  2 bad checksums
  0 incomplete headers
  0 bad data length fields
  2 datagrams dropped (no socket)
  0 socket buffer overflows
  0 broadcast/multicast datagrams dropped (no socket)
  0 unexpected enhanced-UDP datagrams received
  0 input timeout exceeded
```

Return Codes

Here are the exit values for the *oenetstat* command:

Exit Value	Description
0	command completed successfully
non-0	command failed — an error message is displayed

oenetopts—Set Network Options

Purpose

The *oenetopts* command is a REXX EXEC that dynamically sets or displays the current network options. Network options can be set and displayed at the same time. This command operates on the currently running HSAS stack, and must be run again after each system IPL if the default options are not desired.

Format

```

▶▶ oenetopts -a config_pathname ▶▶

```

Usage

This command might typically be issued from the `/etc/rc` script that executes when OpenEdition initializes. Superuser authority is required to set or display the configuration options.

Flags

-a Displays all the configurable options and their current values.

Parameters

config_pathname

specifies the HFS pathname for the file containing the configuration values to be updated. The file contains lines of the form *keyword=value*. The following keywords can be used:

bsd_urgent

specifies whether the urgent data pointer is to follow the BSD rules (= yes) or RFC 1122 rules (=no). The default is yes.

buffers

specifies the maximum number of 32K buffers that can be obtained to satisfy I/O requests. Specify a positive whole (preferably even) number. This number is distributed equally among the read and write flows. buffers are only allocated if there is a demand for them. Specifying a maximum buffer value lower than the current buffer allocation will not cause the current buffer allocation to change. The default number of buffers is 64 for read and 64 for write (buffers=128).

Care must be taken not to supply too low a number, since this could adversely affect performance. You can use the *oenetstat -m* shell command to determine whether current buffer use approaches the maximum.

keepalive_extrabyte

specifies whether an extra garbage byte should be sent (=yes) with a keepalive probe or not (=no). The default is no.

keepalive_idle

specifies the time in minutes a connections must be idle before sending a keepalive probe. The default is 120 minutes and the maximum allowed is 256 minutes.

High Speed Access Services

default_mss

specifies the default maximum segment size in bytes. The default is 512 and the range allowed is from 512 to 32728.

tcp_ttl

specifies the default time to live in hop counts. The default is 60.

tos

specifies the default type of service as a hex byte. The default is 00 (interpreted as X'00')

The configuration file may also contain single, whole line REXX comments (*/*...*/*) or blank lines. Each keyword is processed in the order specified, regardless of whether there are duplicate keywords.

Results

The resulting display shows which options have been processed, and, if *-a* is specified, the current values of all options.

Examples

Use the following command to specify that input from the terminal be used in place of the configuration file:

```
oenetopts /dev/fd0
```

Next, enter one or more parameter followed by the end-of-file sequence (Esc-D).

```
buffers=72  
default_mss=1024  
(Esc-D)
```

The previous command sequence returns the following output:

```
processing BUFFERS ...  
processing DEFAULT_MSS ...  
configuration processing complete
```

Return Codes

Here are the exit values for the *oenetopts* command:

Exit Value	Description
0	command completed successfully
non-0	command failed — an error message is displayed

Messages and Codes

System messages displayed while setting up or using the HSAS stack are identified as either OpenEdition messages (using the BPX prefix) or TCP/IP messages (using the EZZ prefix). See *OS/390 UNIX System Services Messages and Codes* for a complete description of the OpenEdition messages. See *OS/390 eNetwork Communications Server: IP Messages Volumes 1-3* for a complete description of the TCP/IP messages.

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not 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
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
U.S.A.

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 information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

Trademarks

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

ACF/VTAM	IMS/ESA
Advanced Peer-to-Peer Networking	LANStreamer
AD/Cycle	Library Reader
AIX	LPDA
AIX/ESA	MVS/DFP
AnyNet	MVS/ESA
APPN	MVS/SP
AS/400	MVS/XA
AT	NetView
BookManager	Nways
C/370	OfficeVision/MVS
CICS	OS/2
C/MVS	OS/390
Common User Access	PR/SM
CT	PROFS
CUA	PS/2
DATABASE 2	RACF
DB2	RETAIN
DFSMS	RISC System/6000
DFSMSHsm	RS/6000
DFSMS/MVS	S/370
DRDA	SAA
ESA/390	SP
ESCON	SQL/DS
ES/9000	System/360
ES/9370	System/370
EtherStreamer	System/390
Extended Services	SystemView
FFST	TURBOWAYS
First Failure Support Technology	UNIX System Services
GDDM	VM/ESA
Hardware Configuration Definition	VTAM
IBM	3090
IBMLink	
IMS	

The following terms are trademarks of other companies:

ATM is a trademark of Adobe Systems, Incorporated.

BSC is a trademark of BusiSoft Corporation.

CSA is a trademark of Canadian Standards Association.

DCE is a trademark of The Open Software Foundation.

HYPERchannel is a trademark of Network Systems Corporation.

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.

Bibliography

eNetwork Communications Server for OS/390 V2R6 Publications

Following are descriptions of the books in the eNetwork Communications Server for OS/390 V2R6 library. The books are arranged in the following categories:

- Softcopy Information
- Marketing Information
- Planning
- Installation, Resource Definition, and Tuning
- Operation
- Customization
- Writing Application Programs
- Diagnosis
- Messages and Codes
- APPC Application Suite.
- Multiprotocol Transport Networking (MPTN) Architecture publications

The complete set of unlicensed books in this section can be ordered using a single order number, SBOF-7011.

Softcopy Information

IBM Networking Softcopy Collection Kit CD-ROM(SK2T-6012).

The softcopy library contains softcopy versions of the licensed and unlicensed books for eNetwork Communications Server for OS/390 V2R6.

All of the unlicensed and licensed books described in this section are available in softcopy on this CD-ROM. These softcopy files can be read using any of the IBM BookManager READ programs. They can also be read with the IBM Library Reader program shipped on this CD.

The CD also contains softcopy of the unlicensed books of many other products.

Marketing Information

A Networking Overview and the following IBM Networking Previews are available:

- VTAM
- TCP/IP

Ask your IBM marketing representative for more information.

Planning

OS/390 eNetwork Communications Server: SNA Planning and Migration Guide (SC31-8622). This guide helps you upgrade to eNetwork Communications Server for OS/390 V2R6. It includes:

- Installation procedures
- Planning to upgrade
 - Upward and downward compatibility
 - Software and hardware requirements
 - Storage requirements
 - Impacts of new functions and enhancements performed without changes to user interfaces
 - Changes to installation process
- Upgrading user interfaces
 - Changes to start options
 - Changes to buffer pools
 - Changes to definition statements
 - Changes to IBM-supplied default user-definable tables and modules
 - Changes to user-definable table macroinstructions
 - Changes to commands
 - Changes to messages
 - Changes to SNA application programming interface
 - Changes to installation-wide exit routines
 - Changes to control blocks
- Implementing optional functions and enhancements introduced in eNetwork Communications Server for OS/390 V2R6.
 - Overview of each new function and enhancement introduced since VTAM V4R4
 - Pointers to other books in the library where implementation details can be found.

OS/390 eNetwork Communications Server: IP Planning and Migration Guide (SC31-8512). This book is intended to help you plan for TCP/IP

Bibliography

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 TCP/IP.

Installation, Resource Definition, Configuration, and Tuning

Program Directory. This document is shipped with the product tape and explains the steps for installing VTAM and TCP/IP.

OS/390 eNetwork Communications Server: IP Configuration Guide (SC31-8513). This book is for people who want to configure, customize, administer, and maintain TCP/IP. Familiarity with MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

OS/390 eNetwork Communications Server: SNA Network Implementation Guide (SC31-8563). This book presents the major concepts involved in implementing a SNA network, and includes:

- Buffer pools, slowdown, pacing, storage considerations
- Implementation considerations
- Sample major node definitions
- Migration considerations
- Tables and filters
- TSO, VCNS, and other programs that run with VTAM
- Tuning procedures
- VTAM start options.

Use this book in conjunction with the *OS/390 eNetwork Communications Server: SNA Resource Definition Reference*.

OS/390 eNetwork Communications Server: SNA Resource Definition Reference (SC31-8565). This book describes each VTAM definition statement, start option, and macroinstruction for user tables. It also describes NCP definition statements that affect VTAM. The information includes:

- IBM-supplied default tables (logon mode and USS)
- Major node definitions
- User-defined tables and filters
- VTAM start options.

If you are unfamiliar with the major concepts involved in implementing a SNA network, use this

book in conjunction with the *OS/390 eNetwork Communications Server: SNA Network Implementation Guide*.

OS/390 eNetwork Communications Server: SNA Resource Definition Samples (SC31-8566). This book contains sample definitions to help you implement VTAM functions in your networks, and includes sample major node definitions. Use this book in conjunction with the *OS/390 eNetwork Communications Server: SNA Network Implementation Guide* and *OS/390 eNetwork Communications Server: SNA Resource Definition Reference*.

OS/390 eNetwork Communications Server: AnyNet SNA over TCP/IP (SC31-8578). This guide provides information to help you install, configure, use, and diagnose SNA over TCP/IP.

OS/390 eNetwork Communications Server: AnyNet Sockets over SNA (SC31-8577). This guide provides information to help you install, configure, use, and diagnose Sockets over SNA. It also provides information to help you prepare application programs to use sockets over SNA.

Operation

OS/390 eNetwork Communications Server: IP User's Guide (GC31-8514). This book is for people who want to use TCP/IP for data communication activities such as FTP and Telnet. Familiarity with MVS operating system and IBM Time Sharing Option (TSO) is recommended.

OS/390 eNetwork Communications Server: SNA Operation (SC31-8567). This book serves as a reference for programmers and operators requiring detailed information about specific operator commands. The information includes:

- VTAM commands and start options
- Logon manager commands
- DISPLAY output examples (messages received)
- VSCS commands.

OS/390 eNetwork Communications Server: SNA Operation Quick Reference (SX75-0121). This book contains essential information about VTAM operator commands.

OS/390 eNetwork Communications Server: High Speed Access Services User's Guide (GC31-8676). This book is for end users and system administrators who want to use

applications using a High Speed Access Services connection available in CS for OS/390.

Customization

OS/390 eNetwork Communications Server: SNA Customization (LY43-0110). This book enables you to customize VTAM, and includes:

- Communication network management (CNM) routing table
- Logon-interpret routine requirements
- Logon manager installation-wide exit routine for the CLU search exit
- TSO/VTAM installation-wide exit routines
- VTAM installation-wide exit routines:
 - Command verification exit (ISTCMMND)
 - Configuration services XID exit (ISTEXCCS) with description of IBM-supplied default exit
 - Directory services management exit (ISTEXCDM)
 - Generic resource resolution exit (ISTEXCGR)
 - Performance monitor exit (ISTEXCPM)
 - SDDLU exit (ISTEXCSD) with description of IBM-supplied default exit
 - Session accounting exit (ISTAUCAG)
 - Session authorization exit (ISTAUCAT)
 - Session management exit (ISTEXCAA) with example
 - TPRINT processing exit (ISTRAEUE)
 - USERVAR exit (ISTEXCUV) with description of IBM-supplied default exit
 - Virtual route pacing window size calculation exit (ISTPUCWC)
 - Virtual route selection exit (ISTEXCVR).

OS/390 eNetwork Communications Server: IP Network Print Facility (SC31-8522). This book is for system programmers and network administrators who need to prepare their network to route VTAM, JES2, or JES3 printer output to remote printers using TCP/IP.

Writing Application Programs

OS/390 eNetwork Communications Server: IP API Guide (SC31-8516). 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.

OS/390 eNetwork Communications Server: IP CICS Sockets Guide (SC31-8518). 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.

OS/390 eNetwork Communications Server: IP IMS Sockets Guide (SC31-8519). 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.

OS/390 eNetwork Communications Server: IP Programmer's Reference (SC31-8515). 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. Familiarity with the MVS operating system, TCP/IP protocols, and IBM Time Sharing Option (TSO) is recommended.

OS/390 eNetwork Communications Server: SNA Programming (SC31-8573). This book describes how to use VTAM macroinstructions to send data to and receive data from (1) a terminal in either the same or a different domain, or (2) another application program in either the same or a different domain. The information includes:

- API concepts
 - Cryptography
 - RUs and exchanges
 - Session establishment and termination
- BIND area format
- Communication Network Management Interface
- Dictionary of VTAM macroinstructions
- OPEN or CLOSE errors
- Operating system differences
- Program Operator Coding requirements
- RAPI DSECTs and control block mappings (ACB, ADSP, BLENT, CV29, EXLST, MTS, NIB, NIB DEVCHAR, NIB PROC, RH, RPL, RPL RTNCD=FDB2=FDBK=DSECT)
- RAPI global variables
- Vector lists

Bibliography

- RPL-based macroinstructions
- RPL RTNCD,FDB2 codes
- User exit routines.

OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Guide (SC31-8581). This book describes how to use the VTAM LU 6.2 application programming interface for host application programs. This book applies to programs that use only LU 6.2 sessions or that use LU 6.2 sessions along with other session types. (Only LU 6.2 sessions are covered in this book.) The information includes:

- VTAM's implementation of the LU 6.2 architecture
- Design considerations for LU 6.2 application programs
- Negotiating session limits with partner LUs
- BIND image and response
- Allocating and deallocating conversations
- FMH-5 and PIP data
- Conversation states
- Sending and receiving data
- Using high performance data transfer (HPDT)
- Session- and conversation-level security and data encryption
- Register usage
- Sync point services
- LU 6.2 global variables
- Vector lists
- Sense codes for FMH-7 and UNBIND
- RCPRI,RCSEC codes
- User exit routines.

OS/390 eNetwork Communications Server: SNA Programmers LU 6.2 Reference (SC31-8568). This book provides reference material for the VTAM LU 6.2 programming interface for host application programs. The information includes:

- APPCCMD macroinstructions
- Primary and secondary return codes (RCPRI, RCSEC)
- DSECTs
- Examples of using VTAM's LU 6.2 API
- Register usage

OS/390 eNetwork Communications Server: CSM Guide (SC31-8575). This book describes how applications use the communications storage manager. The information includes:

- Creating and deleting buffer pools
- Obtaining and freeing buffers
- Return codes and reason codes
- DSECTs

OS/390 eNetwork Communications Server: CMIP Services and Topology Agent Guide (SC31-8576). This book describes the Common Management Information Protocol (CMIP) programming interface for application programmers to use in coding CMIP application programs. The book provides guide and reference information about CMIP services and the VTAM topology agent and includes the following topics:

- Management information base (MIB) API functions
- CMIP message strings
- Special CMIP message strings
- Read queue exit routine
- Sample CMIP application program
- VTAM resources as CMIP objects
- Naming conventions for objects
- VTAM resources and OSI states
- Attributes to object cross-reference
- ASN.1 syntax for CMIP messages
- GDMO table format
- ACYAPHDH header file.

Diagnosis

OS/390 eNetwork Communications Server: IP Diagnosis (SC31-8521). This book explains how to diagnose TCP/IP problems and how to determine whether a specific problem is in the TCP/IP product code. It explains how to gather information for and describe problems to the IBM Software Support Center.

OS/390 eNetwork Communications Server: SNA Diagnosis (LY43-0079). This book helps you identify a VTAM problem, classify it, and collect information about it before you call the IBM Support Center. The information collected includes traces, dumps, and other problem documentation. Volume 1 includes information on the following:

- Command syntax for running traces and collecting and analyzing dumps
- Procedures for collecting documentation (VTAM, TSO)
- VIT analysis tool
- Channel programs

- Flow diagrams
- Procedures for locating buffer pools
- CPCB operation codes
- Storage and control block ID codes
- Offset names and locations for VTAM buffer pools.

Volume 2 includes information on the following:

- VIT entries
- VTAM internal trace
- FFST dumps and probes

OS/390 eNetwork Communications Server: Data Areas Volume 1 (LY43-0111). This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

OS/390 eNetwork Communications Server: Data Areas Volume 2 (LY43-0112). This book describes VTAM data areas and can be used to read a VTAM dump. It is intended for IBM programming service representatives and customer personnel who are diagnosing problems with VTAM.

Messages and Codes

OS/390 eNetwork Communications Server: SNA Messages (SC31-8569). This book describes the following types of messages and other associated information:

- Messages:
 - ELM messages for logon manager
 - IKT messages for TSO/VTAM
 - IST messages for VTAM network operators
 - ISU messages for sockets-over-SNA
 - IVT messages for the communications storage manager
 - IUT messages
 - USS messages
- Other information that displays in VTAM messages:
 - Command and RU types in VTAM messages
 - Node and ID types in VTAM messages
- Supplemental message-related information:
 - Message additions, deletions, and changes
 - Message flooding prevention
 - Message groups and subgroups

- Message routing and suppression including descriptor codes, routing codes, and suppression levels for ELM, IKT, IST, and ISU messages
- Message text and description formats
- Message text of MSGLVL option messages including general information on the MSGLVL option
- Message text of all VTAM network operator messages including variable field lengths

OS/390 eNetwork Communications Server: IP Messages Volume 1 (SC31-8517). This volume contains TCP/IP messages beginning with EZA.

OS/390 eNetwork Communications Server: IP Messages Volume 2 (SC31-8570). This volume contains TCP/IP messages beginning with EZB.

OS/390 eNetwork Communications Server: IP Messages Volume 3 (SC31-8674). This volume contains TCP/IP messages beginning with EZY, EZZ, and SNM.

OS/390 eNetwork Communications Server: IP and SNA Codes (SC31-8571). This book describes codes and other information that display in CS/390 messages:

- Sense codes including VTAM sense code hints, SNA sense field values for RPL-based macroinstructions, and 3270 SNA and non-SNA device sense fields
- Return codes for macroinstructions including ACB OPEN and CLOSE macroinstruction error fields, RTNCD-FDB2 return code combinations, and LU 6.2 RCPRI-RCSEC return codes
- Data link control (DLC) status codes
- Status codes including resource status and session state codes
- Wait state event codes and IDs
- Abend codes
- ATM network-generated cause and diagnostic codes

APPC Application Suite

OS/390 eNetwork Communications Server: APPC Application Suite User's Guide (GC31-8619). This book documents the end-user interface (concepts, commands, and messages) for the AFTP, ANAME, and APING facilities of the APPC application suite.

Bibliography

Although its primary audience is the end user, administrators and application programmers may also find it useful.

OS/390 eNetwork Communications Server: APPC Application Suite Administration (SC31-8620). This book contains the information that administrators need to configure the APPC application suite and to manage the APING, ANAME, AFTP, and A3270 servers.

OS/390 eNetwork Communications Server: APPC Application Suite Programming (SC31-8621). This book provides the information application programmers need to add the functions of the AFTP and ANAME APIs to their application programs.

Multiprotocol Transport Networking (MPTN) Architecture Publications

Following are selected publications for MPTN:

Networking Blueprint Executive Overview (GC31-7057)

Multiprotocol Transport Networking: Technical Overview (GC31-7073)

Multiprotocol Transport Networking: Formats (GC31-7074)

OS/390 Publications

For information on OS/390 and other products, refer to *OS/390 Information Roadmap* (GC28-1727-03).

Index

A

- activating a network interface 8, 9
- adding a network route 14
- AIX node, I/O configuration 5
- API support 1
- application limitations 4
- assigning an IP address 8

B

- BPXPRMxx parmlib member 4, 6

C

- checksum, network interface 10
- codes 29
- commands
 - oeifconfig 8
 - oenetopts 27
 - oenetstat 22
 - oeping 18
 - oeroute 14
 - summary 7
- configuring a network interface 8
- creating a network route 8

D

- data link control (DLC) 1
- deactivating a network interface 8, 9
- displaying
 - IP address of the network interface 11
 - memory usage 24
 - network interface 8, 9
 - network route 8, 14, 23
 - process identifier 23

E

- echo request 18
- enterprise system connection (ESCON)
 - configuring the network interface 9
 - I/O configuration 5
- environment variable (`_BPXK_INET_FASTPATH`) 4
- ESCON (enterprise system connection)
 - configuring the network interface 9
 - I/O configuration 5
- establishing a network route 8

F

- fast path environment
 - restrictions 4
 - support 3
- firewall support 4

H

- hex flag descriptions 12

I

- I/O configuration, AIX node 5
- Internet Control Message Protocol (ICMP) 18
- IP address, assigning 8

L

- limitations, application 4

M

- maximum transmission unit (MTU) 10
- messages 29
- multicast IP address 4
- multipath channel (MPC) 1

N

- network interface
 - activating 8
 - assigning an IP address 8
 - configuring 8
 - deactivating 8
 - displaying the IP address 11
 - name 5
 - removing 8
- network route
 - adding 14
 - creating 14
 - displaying 8
 - establishing 8
 - removing 8
- network routing table 23
- Notices 31

O

- oeifconfig command 8
- oenetopts command 27
- oenetstat command 22
- oeping command 18
- oeroute command 14
- Open Systems Adapter (OSA)
 - configuration 5
 - configuring the network interface 8
 - port number 11
- OpenEdition
 - parmlib member 4, 6
- OSA (Open Systems Adapter)
 - configuration 5
 - configuring the network interface 8
 - port number 11

P

- parmlib member, BPXPRMxx 4, 6
- process identifier, displaying 23

R

removing

- network interface 8

- network route 8

removing a network interface 9

removing a network route 14

requirements, software 5

restrictions, fast path 4

routes

- adding 14

- removing 14

routing table 8, 23

- adding 14

- removing 14

S

setup requirements 5

subnet mask 11

support

- API 1

- fast path 3

- VTAM 5

T

testing network connectivity 18

transport resource list entry (TRLE) 5

V

VTAM support 5

X

XPG compliance 3, 4

Readers' Comments — We'd Like to Hear from You

**OS/390 eNetwork Communications Server
High Speed Access Services
User's Guide
Version 2 Release 6**

Publication No. GC31-8676-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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.

Name

Address

Company or Organization

Phone No.



Cut or Fold
Along Line

Fold and Tape

Please do not staple

Fold and Tape



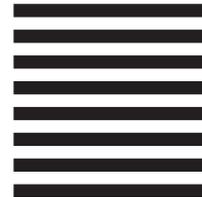
NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Information Development
Department CGMD / Bldg 500
P.O. Box 12195
Research Triangle Park, NC
27709-9990



Fold and Tape

Please do not staple

Fold and Tape

Cut or Fold
Along Line



File Number: S390-50
Program Number: 5647-A01



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

GC31-8676-01

