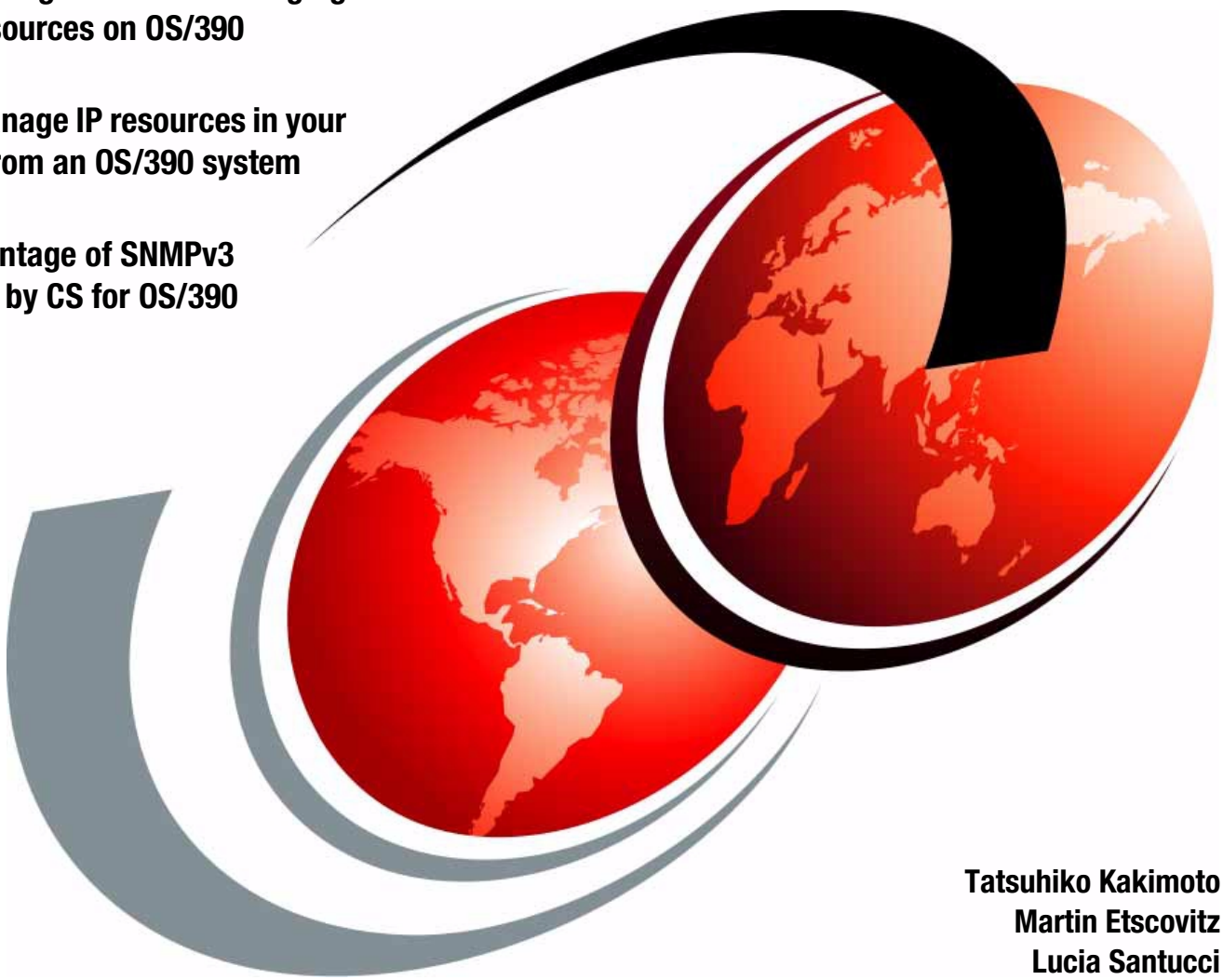
IBM

Managing OS/390 TCP/IP with SNMP

- Sample configuration for managing TCP/IP resources on OS/390
- How to manage IP resources in your network from an OS/390 system
- Take advantage of SNMPv3 supported by CS for OS/390



Tatsuhiko Kakimoto
Martin Etscovitz
Lucia Santucci

Redbooks

ibm.com/redbooks

International Technical Support Organization

SG24-5866-00

Managing OS/390 TCP/IP with SNMP

June 2000

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix F, “Special notices” on page 313.

First Edition (June 2000)

This edition applies to SecureWay Communications Server for OS/390 V2R8 IP Services, Program Number 5647-A01, and Tivoli NetView for OS/390 V1R3, Program Number 5697-B82 for use with OS/390 V2R8.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HZ8 Building 678
P.O. Box 12195
Research Triangle Park, NC 27709-2195

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

Contents

Preface

This redbook addresses the major networking issues facing many large enterprises today: how to manage IP resources on an OS/390 system and how to manage all IP resources in an IP network from an OS/390 system.

System Network Architecture (SNA) was the predominant network architecture for OS/390 systems and as the shift from this hierarchical structure to the heterogeneous peer-to-peer TCP/IP environment takes place, it becomes necessary for the network management infrastructure to adapt to the changing environment. The enterprise of today increasingly relies more on TCP/IP for both the network environment and applications. Effective and timely host management is essential for maintaining reliability and availability.

This redbook provides information that will help a network administrator to implement and configure the Simple Network Management Protocol (SNMP) function of Communications Server for OS/390 IP Services and Tivoli NetView for OS/390.

The SNMP Version 3 (SNMPv3) function of CS for OS/390 provides a secure management environment that allows an OS/390 host to become a managed and/or managing node within the enterprise. Tivoli NetView for OS/390 provides a powerful, centralized interface for managing IP nodes regardless of which platform it is running on. In this redbook you will find many examples from the ITSO environment that will assist in managing OS/390 TCP/IP with SNMP.

The team that wrote this redbook

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

Tatsuhiko Kakimoto is an Advisory Networking Specialist at the International Technical Support Organization, Raleigh Center. He writes extensively and teaches IBM classes worldwide on all areas of networking, especially CS for OS/390 IP Services. Before joining the ITSO in 1999, he worked in the technical support department in IBM Japan designing, implementing IP networks, and consulting major IBM customers.

Martin Etscovitz is a Systems Management Integrator with IBM Global Services in RTP, North Carolina. He has 16 years of experience in S/390 systems supporting VTAM and TCP/IP on OS/390 and VM/ESA. His current responsibilities are supporting TCP/IP on OS/390 and VM/ESA for IBM Global Services Network Outsourcing.

Lucia Santucci joined IBM Italy in 1987 and she worked at customer sites, implementing network management solutions with the NetView family of products. She also worked in the past six years with automation products like AON and AOC helping customers develop system and network management solutions. Today, you can find her at the Customer Support Center in Rome supporting network specific products like VTAM and TCP/IP.

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

Robert Haimowitz
Gail Christensen
Shawn Walsh
International Technical Support Organization, Raleigh Center

Alfred B. Christensen
Sara Haggar
Kristine Adamson
Dinakaran Joseph
IBM CS for OS/390 IP Development, Research Triangle Park, Raleigh, NC

Paul Quigley
Paul Koch
Gregory Smith
Zerbaksh Bam
Haley Gray
Paul M. Smith
Tivoli NetView for OS/390 Development, Research Triangle Park, Raleigh, NC

Comments welcome

Your comments are important to us!

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

- Fax the evaluation form found in “IBM Redbooks review” on page 331 to the fax number shown on the form.
- Use the online evaluation form found at http://www.redbooks.ibm.com/
- Send your comments in an Internet note to redbook@us.ibm.com

Chapter 1. TCP/IP management overview

In the enterprise environment, many of the major S/390 strategic initiatives depend on TCP/IP access to OS/390. Administrative domains are now focusing on Simple Network Management Protocol (SNMP) as a management tool to achieve high reliability and availability within their enterprise. This chapter presents an overview of the TCP/IP network management design philosophy and the network framework available to manage TCP/IP in Communications Server for OS/390 IP Services.

1.1 TCP/IP network management overview

The TCP/IP protocol suite provides easy connectivity and interoperability among several kinds of hosts supplied by different vendors. But since it does not implement a hierarchical approach, it makes the network management process rather different from that to which IBM SNA users are accustomed.

In the early days, managing these interconnected hosts did not seem to be a real issue to TCP/IP users. A small set of tools such as NETSTAT, PING and TRACEROUTE would fulfill most of their management needs. But as networks started to grow in size and complexity, and as more mission critical applications were converted to TCP/IP, the existence of sophisticated management tools to help keep them operational became an important requirement.

In 1987, the Simple Gateway Monitoring Protocol (SGMP) was introduced as a first attempt to standardize the TCP/IP management process. But while it was still being implemented, additional work that was being carried out led to the development of another protocol known as Simple Network Management Protocol (SNMP), which made SGMP obsolete.

SNMP was proposed in 1988 by the Internet Activities Board (IAB) and, in less than two years, many vendors made it available in their TCP/IP implementations. As its name implies, it includes a simple set of requests and responses that are exchanged between network managing stations, which play the role of managers, and network elements that are to be managed, which are usually referred to as SNMP agents.

The information that can be retrieved from the agents resides in the Management Information Base (MIB). It defines the managed objects, such as interface statuses, routings and packet counts that may be used to monitor and control network elements in a TCP/IP-based network.

SNMP is an industry-wide standard for reporting management data. Because different platforms implement the same protocols, management information can be requested and sent across heterogeneous platforms. Someone familiar with retrieving TCP or IP counts from OS/390, for example, can request the same information from other IBM or non-IBM platforms that implement SNMP.

Examples of the use of management data:

- Query IP packet counters to see how many packets are coming in, how many are being sent, and how many are being discarded due to errors.

- Monitor for hardware devices going down so that appropriate action can be taken.
- Drop TCP/IP connections.
- With really smart management applications, IP topology mapping shows where bottlenecks are possible. Such smart applications include those as part of Tivoli NetView for UNIX, Tivoli NetView for Windows NT and a number of non-IBM products, such as HP Open View.

The protocols are defined by the Internet Engineering Task Force (IETF) working groups. Any new protocols (for routing, for example) are expected to also define the management data used to manage the new function. The management data is to be described using SNMP's language and structures.

1.2 Network management framework

Over time, as the network management framework has evolved from SNMPv1, through SNMPv2, to SNMPv3, the definitions of each of these architectural components have become richer and more clearly defined, but the fundamental architecture has remained consistent. This section describes the following three components that the current network management framework for TCP/IP-based internetworks consists of:

- Structure and identification of Management Information (SMI), which defines how managed objects contained in the MIB are defined.
- Management Information Base (MIB), which defines the managed objects in TCP/IP-based networks.
- Simple Network Management Protocol (SNMP), which specifies the protocol used to access the managed objects.

1.2.1 Structure and Identification of Management Information (SMI)

The Structure of Management Information (SMI) defines the rules by which managed objects are named and defined. Managed objects are accessed via a virtual information store, known as the Management Information Base or MIB. Objects in the MIB are defined using a subset of Abstract Syntax Notation One (ASN.1).

Each object type has a name, a syntax and an encoding. The name is represented uniquely as an object identifier, which is an administratively assigned name.

The syntax for an object type defines the abstract data structure corresponding to that object type. For example, the structure of a given object type might be an integer or an octet string. RFC 1155 restricts the ASN.1 constructs which may be used. These restrictions are made solely for the sake of simplicity.

The encoding of an object type is simply how instances of that object type are represented using the object type's syntax. RFC 1155 specifies the use of the basic encoding rules of ASN.1.

1.2.1.1 Names

Names are used to identify managed objects and they are hierarchical in nature. For example, each international standard has an object identifier assigned to it for

the purpose of identifying some object, regardless of the semantics associated with it.

An object identifier is a sequence of integers which traverse a global tree. The tree consists of a root connected to a number of labeled nodes. Each node may, in turn, have children of its own and, in this case, it is called a subtree. This process may continue to an arbitrary level of depth. Each leaf in the tree is a data object. Each data object has a numeric object identifier (such as 1.3.6.1.2.1.1.1) that describes its location in the tree. Objects also have a textual name (such as sysDescr) that translates to the numeric object identifier but provides a more usable way to identify a piece of data. Textual conventions for MIB objects were introduced in SMI Version 2 (SMIv2) and documented in RFC 2579. Some objects (such as a device name) can have multiple copies, and each piece of data is further qualified by an instance identifier. For example, the index for the device name of the sixth interface would be 6, as in ifName.6. For objects for which there is only one instance, the instance identifier is zero, as in sysDescr.0, which is the system description.

The root node itself is unlabeled. Under the iso(1) node, the ISO has designated one subtree for use by other international organizations, org(3). Under this node, one of the subtrees has been allocated to the US Department of Defense, dod(6), under which only one node has been assigned to the Internet community, and it will be administered by the IAB. Therefore, the Internet subtree of object identifiers starts with the prefix 1.3.6.1.

1.2.1.2 The management subtree

Under the IAB node, four subtrees have been defined, namely, directory(1), mgmt(2), experimental(3) and private(4). The administration of the mgmt(2) subtree was delegated by the IAB to the Internet Assigned Numbers Authority for the Internet. This subtree is used to identify objects which are defined in IAB approved documents. As RFCs, which define new versions of the Internet standard MIB, are approved they are assigned an object identifier for identifying the objects defined by that memo. The Internet standard MIB has been assigned management document number one. Therefore, its object identifier is {mgmt 1} or 1.3.6.1.2.1.

The private(4) subtree is also of importance to the management tasks. It is used to define enterprise specific variables in the MIB, which means that users can add their own objects to the MIB. The prefix of these variables will be 1.3.6.1.4.

Figure 1 on page 8 illustrates the hierarchical ISO tree that has been adopted to identify managed objects.

The SMI is not supposed to define objects in the MIB, but it does specify a format to be used by the RFCs which define these objects. An object type definition consists of five fields as shown below:

- Object: a textual name, termed the object descriptor, for the object type, along with its corresponding object identifier.
- Syntax: the abstract syntax for the object type, such as integer, octet string, counter, timeticks and so forth.

- Definition: a textual description of the semantics of the object type.
- Access: one of read-only, read-write, read-create, write-only or not-accessible.
- Status: one of current, depreciated or obsolete.

Following is an example of an object type definition:

```
Object:
   sysDescr.
Syntax:
   DisplayString (SIZE (0..255))
Definition:
   A textual description of the entity. This value should include the full
   name of the system's hardware type, software operating system, and
   networking software. It is mandatory that this only contain printable
   ASCII characters.
Access:
   read-only.
Status:
   current.
```

Further information on this subject can be obtained in RFC 1155.

1.2.2 Management Information Base (MIB)

The Management Information Base (MIB) defines the objects that may be managed for each layer in the TCP/IP protocol. There are two versions: MIB-I and MIB-II. MIB-I was defined in RFC 1156, and is now classified as a historic protocol with a status of not recommended.

Each managed node supports only those groups that are appropriate. For example, if there is no gateway, the EGP group need not be supported. But if a group is appropriate, all objects in that group must be supported. The list of managed objects defined has been derived from those elements considered essential. This approach of taking only the essential objects is not restrictive, since the SMI provides extensibility mechanisms such as definition of a new version of the MIB and definition of private or nonstandard objects.Objects are, for example, packet counters, routing tables and outage information and are arranged into groups and each managed node should support only those groups that are appropriate. Below are some examples of objects in each group. The complete list is defined in RFC 1213. Please also refer to RFC 2011, RFC 2012 and RFC 2013 for updated information of IP, TCP and UDP.

A document that identifies the characteristics of particular SNMP implementations was introduced with SNMPv2, namely capability statements. Capability statements describe an agent's capabilities with respect to object and event notification definitions. The capability statements of the SNMP agent and subagent shipped with CS for OS/390 V2R8 IP can be found as /usr/lpp/tcpip/samples/mvstcpip.caps.

Below are not the complete MIB definitions but presented as an example of the objects defined in each group.

Table 1. Management Information Base II (MIB-II). Group definition

Group	Objects for	#
System	Basic system information	7

Group	Objects for	#
Interfaces	Network attachments	23
AT	Address translation	3
IP	Internet protocol	38
ICMP	Internet control message protocol	26
TCP	Transmission control protocol	19
UDP	User datagram protocol	7
EGP	Exterior gateway protocol	18
SNMP	SNMP application entities	30
# = Number of objects in the group		

- System: this group contains information about the entity, such as system hardware, software and its version:
  - sysDescr - Full description of the system (version, HW, OS)
  - sysObjectID - Vendor's object identification
  - sysUpTime - Time since last re-initialization of SNMP agent
  - sysContact - Name of contact person
  - sysServices - Services offered by device
- Interfaces: this group contains all the interfaces through which the node can send/receive IP datagrams. It also contains counters for packets sent/received and errors:
  - ifIndex - Interface number
  - ifDescr - Interface description
  - ifType - Interface type
  - ifMtu - Size of the largest IP datagram
  - ifAdminiStatus - Status of the interface
  - ifLastChange - Time the interface entered in the current status
  - ifInErrors - Number of inbound packets that contained errors
  - ifOutDiscards - Number of outbound packets discarded
- IP: this group contains information about the IP layer, such as the number of datagrams sent, received and forwarded. It includes a table object, the IP address table which maintains the IP addresses associated with different devices. The ipRouteTable defined by MIB-II (RFC 1213) is considered obsolete and is replaced with the ipForwardTable defined by RFC 2096, which defines objects for managing routes in the IP environment:
  - ipForwarding - Indication of whether this entity is an IP gateway
  - ipInHdrErrors - Number of input datagrams discarded due to errors in their IP headers
  - ipInAddrErrors - Number of input datagrams discarded due to errors in their IP address

    - ipInUnknownProtos - Number of input datagrams discarded due to unknown or unsupported protocol
    - ipReasmOKs - Number of IP datagrams successfully re-assembled
- ICMP: this group contains the ICMP input and output statistics:
    - icmpInMsgs - Number of ICMP messages received
    - icmpInDestUnreachs - Number of ICMP destination-unreachable messages received
    - icmpInTimeExcds - Number of ICMP time-exceeded messages received
    - icmpInSrcQuenchs - Number of ICMP source-quench messages received
    - icmpOutErrors - Number of ICMP messages not sent due to problems within ICMP
- TCP: this group contains information about the TCP connections, such as the maximum number of connections the entity can support, the total number of retransmitted segments, the minimum and maximum timeout values and so forth:
    - tcpConnTable - Maintains information about the existing TCP connections
    - tcpRtoAlgorithm - Algorithm to determine the timeout for retransmitting unacknowledged octets
    - tcpMaxConn - Limit on the number of TCP connections the entity can support
    - tcpActiveOpens - Number of times TCP connections have made a direct transition to the SYN-SENT state from the CLOSED state
    - tcpInSegs - Number of segments received, including those received in error
    - tcpConnRemAddress - The remote IP address for this TCP connection
    - tcpInErrs - Number of segments discarded due to format error
    - tcpOutRsts - Number of resets generated
- UDP: this group contains information about the UDP layer, such as counters and datagrams sent and received:
    - udpTable: Maintains information about UDP ports for which local applications are listening datagrams.
    - udpInDatagrams - Number of UDP datagrams delivered to UDP users
    - udpNoPorts - Number of received UDP datagrams for which there was no application at the destination port
    - udpInErrors - Number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port
    - udpOutDatagrams - Number of UDP datagrams sent from this entity
- EGP: this group contains information about EGP peers, such as the number of messages sent and received, and error counts:
    - egpInMsgs - Number of EGP messages received without error
    - egpInErrors - Number of EGP messages with errors
    - egpOutMsgs - Number of locally generated EGP messages

    - egpNeighAddr - The IP address of this entry's EGP neighbor
    - egpNeighState - The EGP state of the local system with respect to this entry's EGP neighbor
- SNMP: this group contains information about the SNMP agent. Some examples are the number of SNMP packets received, the number of SNMP requests and number of requests receiving authentication failures:
    - snmpInPkts - Total number of SNMP messages which were delivered to the SNMP entity and were for a supported SNMP version
    - snmpInBadCommunityNames - Total number of SNMP messages delivered to the SNMP entity which used an SNMP community name not known to said entity
    - snmpEnableAuthenTraps - Indicates whether the SNMP entity is permitted to generate authenticationFailure traps

The system group is the first group in the MIB. Therefore, it is identified as:

```
{ mib 1 }
```

or

```
1.3.6.1.2.1.1
```

Each group has its own subtree. For example, the first variable in the system group is the system description (sysDescr), which can be represented as:

```
{ system 1 }
```

or

```
1.3.6.1.2.1.1.1
```

To summarize the explanation, the composition of the ASN.1 object identifier for sysDescr is shown below:

iso	org	dod	internet	mgmt	mib	system	sysDescr
1	3	6	1	2	1	1	1

Figure 1 illustrates some managed MIB objects in a hierarchical tree structure. An object’s location in the MIB tree sytematically identifies how to access this object. Note that the rules for naming and defining these objects are specified by SMI.

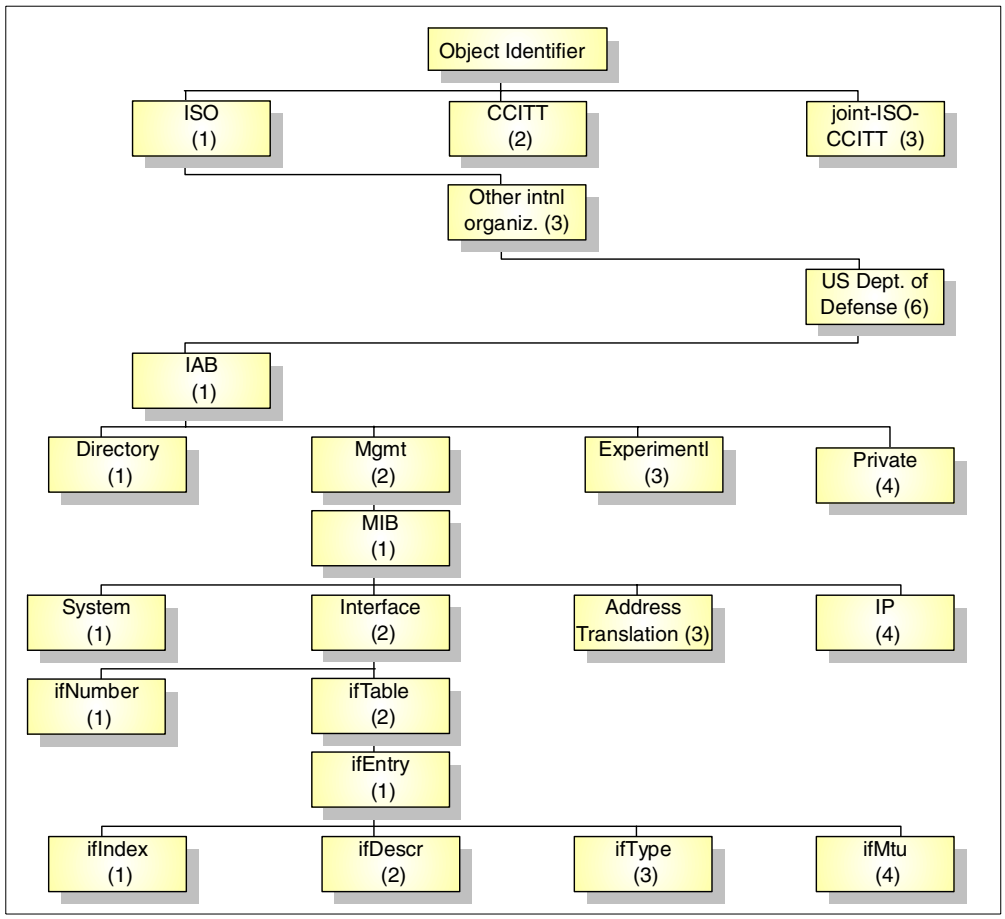


Figure 1. MIB-II: Object identifier allocation for TCP/IP-based network

1.2.3 Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) allows for architected management type actions and information. The SNMP defines the protocols which are used to manage the objects defined in the MIB. The separation of definition of the managed objects and the protocols used to monitor them brings some advantages, such as:

- New objects may be added without a need to change the protocols.
- Multiple protocols can manage the same objects.

There are three versions of SNMP available, usually referred as SNMPv1, SNMPv2, and SNMPv3 respectively. Considering the security functions provided by the SNMP protocols, it is categorized into the following two levels:

- Community-based security model, whose data is protected by nothing more than a password, namely the community name. This level of security is provided by the SNMPv1 and SNMPv2c Community-Based Security Model.
- User-based security model (USM), which provides different levels of the security based on the user accessing the managed information. To support this security level, the SNMPv3 framework defines several security functions, such as USM for authentication and privacy, and view-based access control model (VACM) which provides the ability to limit access to different MIB

objects on a per-user basis, and the use of authentication and data encryption for privacy.

SNMPv3 uses the same protocol operations that were defined for SNMPv1 and SNMPv2. Because the SNMPv3 framework includes the support of both SNMPv1 and SNMPv2, existing SNMP manager applications may be used to manage the network information from devices that support SNMPv3. A manager or agent could choose to implement only SNMPv3. However, practically speaking, it is unlikely it would not support at least SNMPv1.

1.2.3.1 SNMP network management components

The network management framework for TCP/IP-based internetworks consists of the following entities:

- SNMP managers

  An SNMP manager is an application at a network management host that typically requests the management data (the GET and SET requests) from an agent using the SNMP protocol.

- SNMP agents

  An SNMP agent is a server at a managed host which responds to the SNMP requests from managers. An agent must be present at each IP host in order to enable management of that host by an SNMP manager. An SNMP agent is basically a server that provides support for the Management Information Base (MIB) objects that are pertinent to the IP host at which it resides.

- SNMP subagents

  Commonly, an SNMP agent does not maintain the values of all MIB objects directly. These values are generally maintained or accessed through one or more subagents. An SNMP subagent interacts with an agent to respond to the GET and SET requests for a particular set of objects it owns.

Managers and agents do not typically have a one-to-one relationship. A network management application can be requesting data from many different agents. Likewise, an agent can be communicating data to many different managers.

SNMP is used as a communication protocol to exchange information between SNMP agents and managers. The interface used to exchange information between the CS for OS/390 IP SNMP agent and SNMP subagents is the Distributed Protocol Interface (DPI), which is defined in RFC 1592.

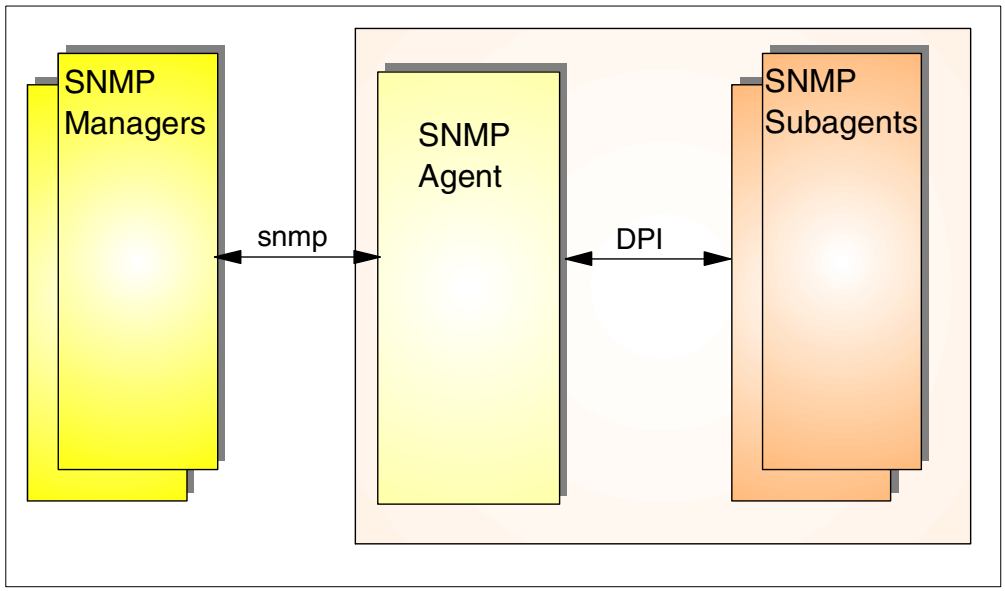


Figure 2. Functional components of SNMP

RFC 1157 defines the Network Management Station (NMS) as the one that executes network management applications (NMA) that monitor and control network elements (NE) such as hosts, gateways and terminal servers. These network elements use a management agent (MA) to perform the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements.

All the management agent functions are only alterations (set) or inspections (get) of variables limiting the number of essential management functions to two and avoiding more complex protocols. In the other direction, from NE to NMS, a limited number of unsolicited messages (notifications) are used to indicate asynchronous events. In the same way, trying to preserve the simplicity, the interchange of information requires only an unreliable datagram service and every message is entirely and independently represented by a single transport datagram. This means also that the mechanisms of SNMP are generally suitable for use with a wide variety of transport services. RFC 1157 specifies the exchange of messages via the UDP protocol, but a wide variety of transport protocols can be used.

The entities residing at management stations and network elements that communicate with one another using SNMP are termed SNMP application entities. The peer processes that implement it are the protocol entities. An SNMP agent with some arbitrary set of SNMP application entities is called an SNMP community, where each one is named by a string of octets that need to be unique only to the agent participating in the community.

1.2.3.2 SNMP versions

The overview of SNMP versions is shown below:

SNMPv1

SNMPv1 is defined in RFC 1155 and RFC 1157. SNMPv1 provides community-based security.

A message in the SNMPv1 protocol consists of a version identifier, an SNMP community name and a protocol data unit (PDU). It is mandatory that all implementations of SNMPv1 support the five PDUs:

- GetRequest: retrieve the values of a specific object from the MIB.
- GetNextRequest: retrieve the next logical management data object(s) in the MIB tree.
- SetRequest: alter the values of a specific object from the MIB.
- GetResponse: response from a GetRequest, a GetNextRequest or a SetRequest.
- Trap: Capability of the network elements to generate events to network management stations such as agent initialization, agent restart and link failure. There are seven trap types defined in RFC 1157: coldStart, warmStart, linkDown, linkUp, authenticationFailure, egpNeighborLoss and enterpriseSpecific.

Figure 3 summarizes the five operations defined by the SNMPv1 framework.

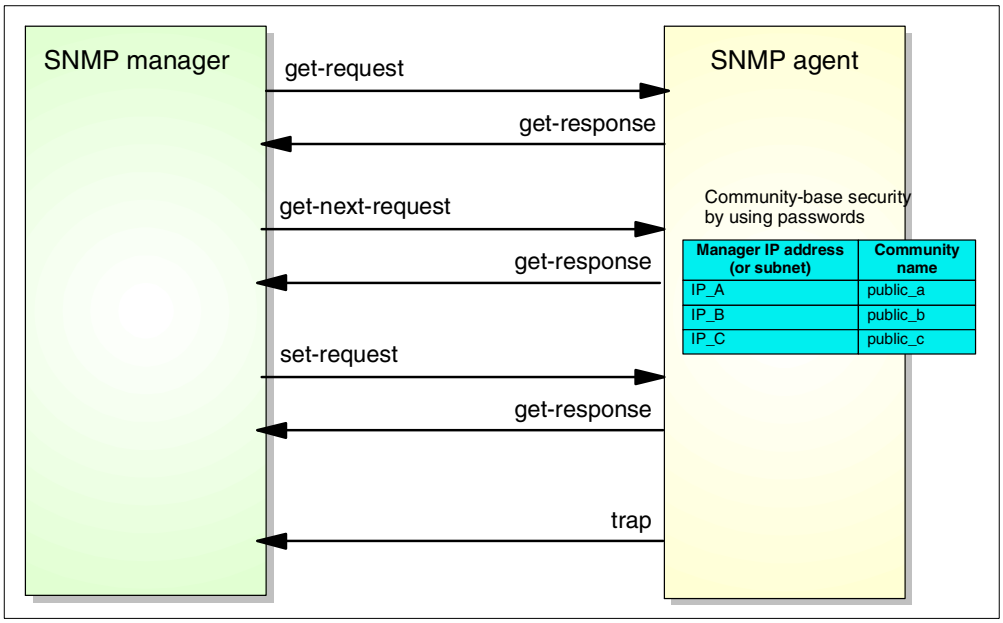


Figure 3. SNMPv1 operations

SNMPv1 supports the community-based security only, which requires that an agent be configured with community names (or passwords) that can be used on requests coming from designated IP addresses. If the community name in a received message matches the community name configured for the IP address from which the message is sent, the incoming request is processed.

RFC 1157 defines six specific traps, with the seventh one allowing a vender to implement an enterprise-specific trap as shown in Table 2.

Table 2. SNMP traps

Trap type	Trap name	Description
0	coldStart	SNMP agent is reinitializing itself such that the agent's configuration or the protocol entity implementation may be altered.
1	warmStart	SNMP agent is reinitializing itself such that neither the agent configuration nor the protocol entity implementation is altered.
2	linkDown	SNMP agent recognizes a failure in one of its communication links. The first variable in the message identifies the interface.
3	linkUp	SNMP agent recognizes that one of the communication links represented in the agent's configuration has come up.
4	authenticationFailure	A message was received from an SNMP manager with an invalid community name.
5	egpNeighborLoss	An EGP peer has been marked as down and the peer relationship no longer obtains.
6	enterpriseSpecific	SNMP agent recognizes that some enterprise-specific event has occurred. The specific-trap field identifies the particular event which occurred.

SNMPv2
SNMPv2 adds additional SNMP operations such as GETBULK, provides more granular error responses, introduces the inform-request PDU, and defines new abstract data types (textual convention) and an enhanced MIB definition, called SMIv2.

The SNMPv2 protocol and MIB structure are defined in RFC 1902 through RFC 1908, which in August 1993 became a proposed standard with the status of elective. There are two choices of security defined by RFCs with SNMPv2. One is named SNMPv2c, which supports community-based security like SNMPv1, and the other is SNMPv2u, which provides user-based security and allows installations to define user security names, authentication, and privacy keys. SNMPv2u was defined as an experimental protocol and became a precursor to SNMPv3.

Note: In April 1999 RFCs 1902 through 1903 were obsoleted by RFCs 2578 through 2580.

SNMPv3
SNMP Version 3 (SNMPv3), defined in RFCs 2570 through 2575 issued in April 1999, is the standards-based solution to the previous weaknesses of SNMP security. However, SNMPv3 is not just enhanced security. Actually, it defines an architecture for SNMP management frameworks, with the intent that pieces of the architecture can advance over time without requiring the entire structure to be rewritten.

For that reason, three major subsystems were defined as shown below. The framework is structured so that multiple models can be supported concurrently and replaced over time. For example, although there is a new message format for SNMPv3, messages created with the SNMPv1 and SNMPv2 formats can still be supported. Similarly, the user-based security model can be supported concurrently with the community-based security models previously used.

- Message processing subsystem

  This subsystem allows support of SNMPv1 and SNMPv2 protocol data units concurrently while supporting SNMPv3 protocol data units.

- Security subsystem

  In this subsystem, community-based and user-based security may be supported concurrently. The following security services are also provided by the user-based security model:

  a. Authentication of messages

  b. Encryption/decryption messages for privacy.

- Access control subsystem

  This susbsystem determines whether access to managed information should be allowed. Currently only the view-based access control model (VACM) has been defined.

Figure 4 illustrates the structure of an SNMP entity defined in the SNMPv3 framework. In this framework, an SNMP entity is one that we used to call an SNMP agent and SNMP manager and is composed of two elements: an SNMP engine and SNMP application.

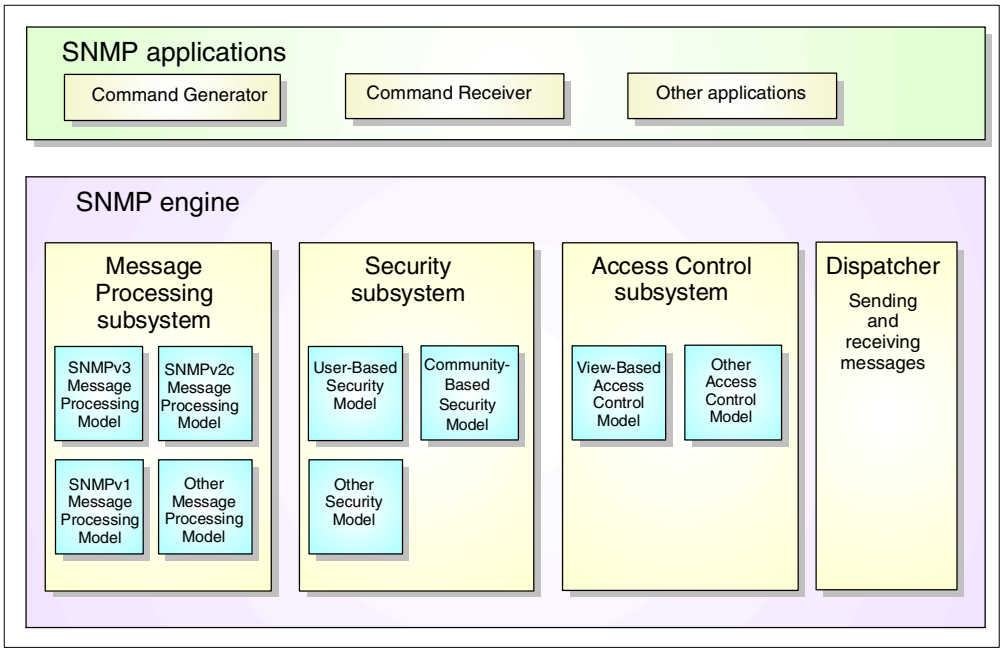


Figure 4. SNMP entity structure

RFC 2574 describes the user-based security model (USM) for message security. USM uses a basic concept of a user, on whose behalf SNMP messages are

generated. For USM to work, the user must be defined to both the manager and the agent. For an authenticated request on behalf of the user, both the manager and the agent must know a set of one or more "secrets" (keys) to be used in processing the message. These keys are the authentication key and optionally, the privacy key.

The authentication protocols currently supported with USM are:

- HMAC-Message Digest 5 , or HMAC-MD5, which uses a 128-bit key (16 bytes)
- HMAC-Secure Hash Algorithm, or HMAC-SHA, which uses a 160-bit key (20 bytes)

The privacy protocol supported with USM is Cipher Block Chaining Data Encryption Standard, or CBC 56-bit DES. All of the protocols are based on symmetric cryptography (private key) mechanisms.

Defined in RFC 2575, the View-based Access Control Model is used to indicate who can use which management data and in what ways. With VACM, one defines views of management data. They can be broad views (like *everything in the MIB tree*) or very specific (like *only instance number 2 of all the fields in the ifTable*). Then a group is defined to indicate a set of users whose access rights are the same. Finally, the access rights are defined to match groups with the views they may use. Views can be defined for read access, for write access, and for notification access. By configuring the users' view for notification access, you can define what data can be sent to users specified using trap or inform messages.

The message processing for SNMPv3 PDUs and the SNMPv3 message format are defined in RFC 2572. In this specification, the message header format unique to SNMPv3 is defined. However, the data portion of the SNMPv3 PDU has been defined to use one of those specified in RFC 1905 that defines the SNMPv2 PDU format. As a trap PDU over the SNMPv3 protocol, the SNMPv2 trap PDU is used too.

SNMPv3 provides the ability to configure the agent dynamically, from either a local or remote host, and to make changes in the configuration while the SNMP agent is running. Doing SNMP agent configuration dynamically requires a good understanding of how the SNMP SET commands can be issued to create new rows or to change or delete existing rows, and a familiarity with the SNMP configuration tables defined in RFCs 2570 through 2575.

1.2.3.3 Related RFC documents

The following is a list of RFC documents that define each SNMP version:

SNMPv1

The following RFCs define SNMPv1:

- *RFC 1155 - Structure and identification of management information for TCP/IP-based internets*, which defines SMIv1, the mechanisms used for describing and naming objects for the purpose of management.
- *RFC 1157 - Simple Network Management Protocol (SNMP).*
- *RFC 1212 - Concise MIB definitions*, which defines a more concise description mechanism, and is wholly consistent with the SMIv1.

- *RFC 1213 - Management Information Base for Network Management of TCP/IP-based internets:MIB-IISNMPv2*, which defines the base set of MIB-II objects.
- *RFC 1215 - Convention for defining traps for use with the SNMP*, which defines a convention for defining Traps for use with SMIv1.

SNMPv2

SNMPv2 is defined by these documents:

- *RFC 1905 - Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*.
- *RFC 1906 - Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2).*
- *RFC 1907 - Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2).*
- *RFC 2578 - Structure of Management Information Version 2 (SMIv2)*.
- *RFC 2579 - Textual Conventions for SMIv2.*
- *RFC 2580 - Conformance Statements for SMIv2*, which defines conformance statements and requirements for defining agent and manager capabilities.

The following document increases the definition of SNMPv2:

- *RFC 1901 - Introduction to Community-based SNMPv2*, which is an experimental definition for using SNMPv2 PDUs within a community-based message wrapper, and is usually referred to as SNMPv2c.

SNMPv3

SNMPv3 is defined by these documents:

- *RFC 2570 - Introduction to Version 3 of the Internet-standard Network Management Framework*, which describes the SNMPv3 framework.
- *RFC 2571 - An Architecture for Describing SNMP Management Frameworks*
- *RFC 2572 - Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
- *RFC 2573 - SNMP Applications*
- *RFC 2574 - User-based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*, defines USM providing for both authenticated and encrypted SNMP messages.
- *RFC 2575 - View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP).* VACM is defined by this document to provide the ability to limit access to different MIB objects on a per-user basis.

SNMPv3 also uses the SNMPv2 definitions of RFCs 1905 through 1907 and the SMIv2 definitions of 2578 through 2580 described above.

Also refer to *RFC 2576 - Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework* for information on coexistence between three versions of the network management framework.

Note: The RFC lists above are not a complete list of RFCs that define the SNMP related standards. For a complete list of the RFC documents, please refer to the most recent rfc-index.txt file, which can be found in:

```
ftp://ftp.isi.edu/in-notes/rfc-index.txt
```

1.3 Network management implementation on the OS/390 environment

Communications Server for OS/390 IP Services provides management services for the TCP/IP stack including SNMP managers, SNMP agents and subagents. Tivoli NetView for OS/390 also can communicate as an SNMP manager with OS/390 SNMP agents and subagents. The following sections describe the type of management data supported by the CS for OS/390 SNMP agent and subagents and how this data can be used to support network management as well as the SNMP management functions included in Tivoli NetView for OS/390.

1.3.1 Communications Server for OS/390 IP Services

The UNIX SNMP manager (the `osnmp` command) and SNMP agent shipped as part of CS for OS/390 V2R8 IP support all three versions of the SNMP protocol. Note, however, that the NetView SNMP command supports SNMPv1 only. SNMPv2 introduced two choices of security: SNMPv2c which provides community-based security and SNMPv2u which provides user-based security. CS for OS/390 V2R8 IP supports SNMPv2c but not SNMPv2u. CS for OS/390 did support SNMPv2u in V2R5 and V2R6. Its support was dropped in V2R7 because the standards-based SNMPv3 support was introduced. Table 3 shows which levels of security are supported on the different TCP/IP releases.

Table 3. Security support by TCP/IP release

	SNMPv1	**SNMPv2c**	**SNMPv2u (USEC)**	**SNMPv3 (USM & VACM)**
IBM TCP/IP for MVS V3R2	NetView SNMP command, MVS SNMPD agent	not supported	not supported	not supported
OS/390 V2R4 OpenEdition TCP/IP	osnmp command, OSNMPD agent	osnmp command, OSNMPD agent	osnmp command	not supported
OS/390 V2R5	NetView SNMP command, osnmp command, OSNMPD agent	osnmp command, OSNMPD agent	osnmp command, OSNMPD agent	not supported
OS/390 V2R6	NetView SNMP command, osnmp command, OSNMPD agent	osnmp command, OSNMPD agent	osnmp command, OSNMPD agent	not supported
OS/390 V2R7 and above	NetView SNMP command, osnmp command, OSNMPD agent	osnmp command, OSNMPD agent	not supported	osnmp command, OSNMPD agent

The SNMP manager can be the `osnmp` command, the NetView SNMP command or any other manager in a TCP/IP network. SNMP commands such as the `osnmp` command and the NetView `SNMP` command can be used to access MIB object

information. They send SNMP requests to SNMP agents on either local or remote hosts.

Note: Since CS for OS/390 V2R8 IP, the `snmp` command that can be issued in the UNIX System Services shell environment is provided. This command is a synonym for the `osnmp` command.

The CS for OS/390 SNMP agent supports the Distributed Protocol Interface (DPI), by which subagents can communicate with the CS for OS/390 SNMP agent to support particular MIB objects. For more information about DPI, refer to *OS/390 SecureWay Communications Server IP Programmer's Reference*, SC31-8515.

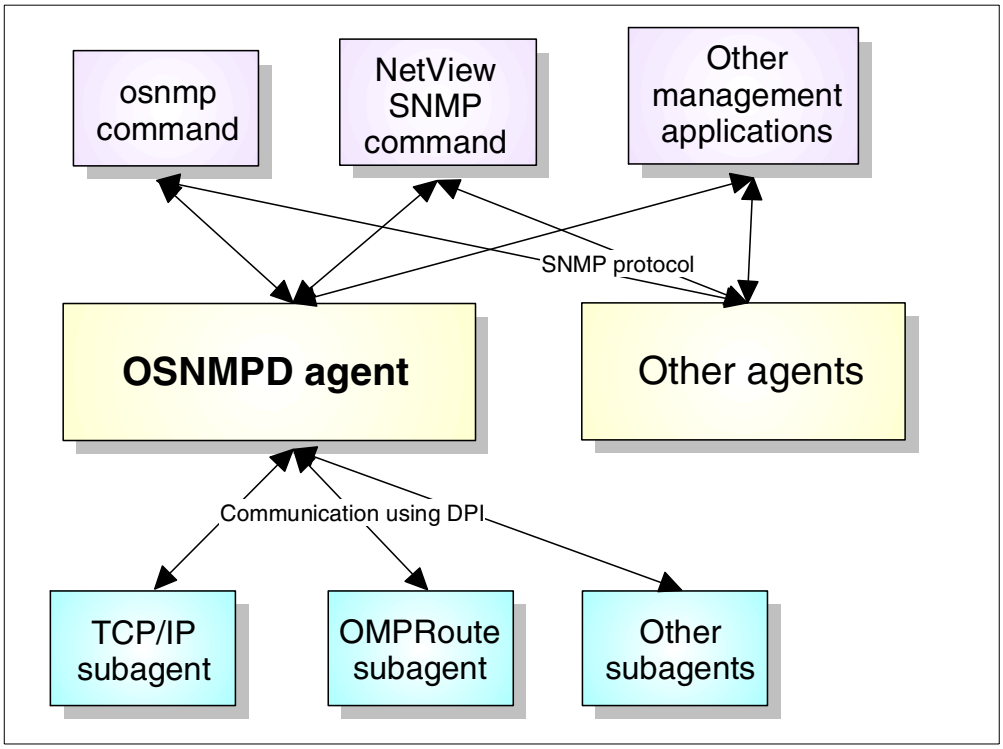


Figure 5. Functional components of SNMP

Encryption is an optional feature of USM, and available as a separately orderable encryption feature where export laws allow. See Table 4 for the list of security features available since CS for OS/390 V2R7 IP. Encryption features are not priced and must be ordered separately from the OS/390 base. Each encryption feature is packaged so that one feature cannot overlay another.

Table 4. Encryption features

Encryption Feature	OS/390 V2R8	OS/390 V2R7	SNMP Elements
Level 1	JTCP383	JTCP373	No SNMP privacy supported
Level 2	JTCP382	JTCP372	SNMP CBC 56-bit DES
Level 3	JTCP38K	JTCP37K	SNMP CBC 56-bit DES

The agent and subagents support many MIB objects defined as standard objects in RFCs. Additionally, the SNMP agent and the TCP/IP subagent support nonstandard MIB objects, called enterprise-specific objects. The complete list of MIB objects supported by the SNMP agent and subagents is in Appendix C, “Management Information Base (MIB) objects” on page 213.

The following three SNMP subagents are shipped with CS for OS/390 V2R8 IP:

The TCP/IP subagent reports information maintained in the TCP/IP stack as well as the information related to the ATM OSA-2 adapter. The TCP/IP subagent can retrieve ATM OSA-2 data from the Open Systems Adapter Support Facility (OSA/SF) for support of ATM management. In addition to the standard MIB objects, the TCP/IP subagent provides extensive support for enterprise-specific MIB objects to facilitate:

- Retrieval of response time data between two remote hosts
- Retrieval of TCP/IP configuration, interface and TCP/UDP connection data
- Support for the IBM 3172 Interconnect controller
- Support for ATM management

The OMPROUTE subagent reports the information specific to OSPF. The OMPROUTE subagent provides an alternative to DISPLAY commands for displaying Open Shortest Path First (OSPF) protocol configuration and state information.

The Service Level Agreement (SLA) subagent reports information about defined service policies and performance statistics related to traffic using those policies.The SLA subagent allows network administrators to retrieve data and determine if the current set of SLA policy definitions are performing as needed or if adjustments need to be made. Unlike the TCP/IP and OMPROUTE subagents, the SLA subagent requires the Policy agent to be running as a separate started task.

Table 5 lists SNMP components in different releases of CS for OS/390 IP.

Table 5. SNMP components on OS/390

	OS/390 V2R4 1	OS/390 V2R5	OS/390 V2R6	OS/390 V2R7	OS/390 V2R8
UNIX SNMP manager	X	X	X	X	X 2
UNIX SNMP agent	X	X	X	X	X
TCP/IP subagent	X	X	X	X	X
OMPROUTE subagent				X	X
SLA subagent					X

1 OS/390 TCP/IP OpenEdition
2 the `snmp` command, which is a synonym for osnmp, is provided in OS/390 V2R8

1.3.2 Tivoli NetView for OS/390

Tivoli NetView for OS/390 V1R3 greatly extends its management capabilities in the TCP/IP environment.

You can use Tivoli NetView for OS/390 to assist you with management of your TCP/IP resources by proactively monitoring critical resources and by responding to network events.

In conjunction with CS for OS/390 IP, any TCP/IP or UNIX command, such as `ping`, `tracerte`, or `netstat`, can be issued from NetView. You can also display the status of TCP connections such as TN3270, FTP, and SMTP, and then diagnosis problems for those connections. You can also manage IP resources and NetView TSO and UNIX servers. The proactive monitoring capabilities exploit SNMP functions for MIB polling, for example, identify a failed interface, check MIB thresholding, or check customer-defined performance MIBs for key routers.

Management of TCP/IP resources and sessions attached to CS for OS/390 IP eliminates the need for establishing separate sessions for TSO and UNIX. All management can be centralized in NetView.

Support is also provided for multiple concurrent TCP/IP stacks and for remote communications to other NetView domains and their TCP/IP stacks.

NetView for OS/390 has management advantages other than centralization. The session management function helps you correlate telnet data with SNA data, making diagnostics easier.

Tivoli NetView for OS/390 also supports platform independent Java-based operator interfaces, namely NetView Management Console (NMC), which provides graphical views of your network and system resources, such as network topology and status dynamically discovered and updated by NetView. NMC is a client/server application replacing OS/2-based NetView Graphic Monitor Facility (NGMF).

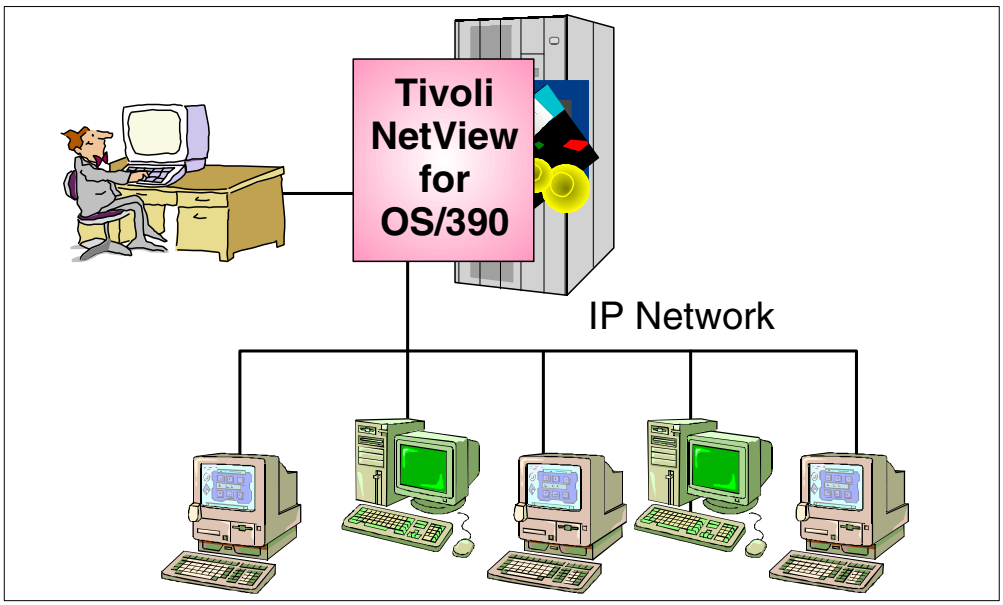


Figure 6. IP managing with NetView V1R3

1.3.2.1 Release comparison

The following table shows the major components and functions of NetView regarding TCP/IP management and when these functions were added or enhanced.

Table 6. Release Comparison

Functional Area	NetView for OS/390		
	V1R1	V1R2	V1R3
Automation			
SNMP Support	-	-	N
Graphical Management			
Platform Independence (Java NMC)	-	N	e
MSM IP Feature	N	e	e
Network Management			
TCP/IP Support	-	N	E
TMR Support	-	N	E
Productivity			
e-mail Support *	-	-	N
NetView 3270 Management Console	N	e	e
Access from Web Browser *	-	N	e
Legend: N = New E = Major enhancements e = enhanced * = This function will not be covered in this redbook			

The following is a brief description of the TCP/IP network management functions supported by Tivoli NetView for OS/390 V1R3 that are relevant for this redbook:

TCP/IP network management

- Graphical network management: It provides graphical views of IP resources in your network and commands to manage them. MultiSystem Manager IP agents are provided on Tivoli NetView for AIX, Tivoli NetView for Windows NT, Tivoli NetView for Sun Solaris, HP OpenView on HP-UX, and HP OpenView on Sun Solaris.
- IP layer 3 discovery: OS/390 UNIX Java agent dynamically discovers IP resources in the managed network and minimizes the need to define resources to be managed. MSM IP agent discovers additional resources, such as IP segments, and the connections from subnets/segments to routers, bridges, and hubs.
- IP, SNMP and UNIX command support: In general, any OS/390 UNIX command can be issued directly from the NetView command line, and can also be used in REXX procedures and other automation routines. The SNMP manager commands that can be issued from NetView panels are:
  - `osnmp`, which is shipped with CS for OS/390 IP and supports all three versions of the SNMP protocol.

    - `jsnmp`, which is provided by Tivoli NetView for OS/390 - a Java-based program running in the OS/390 UNIX environment.
- MIB services: NetView provides a MIB compiler/loader function that allows NetView to support any vendor-specific MIB described using the ASN.1 format. The `loadmib` command enables authorized users to load MIBs from OS/390 UNIX services.
- Pro-active monitoring: CS for OS/390 IP resources can be monitored via one policy definition statement. User exits are provided for both MIB polling and MIB thresholding to allow for more extensive analysis or automation. When defining resources to be monitored, users now have a choice of whether to use PING (simply to determine whether a resource is active) or SNMP functions (to perform MIB polling and thresholding, to define MIB variables to be checked, and check administrative status vs. operational status).
- MIB grouping: NetView has predefined MIB group definitions to collect SNMP data that is most meaningful to operators. Customers can also define groups to implement their own management policies.
- Session management:
    - Support for any socket connection. Session status can be displayed for any socket connection. If the socket is down, automation can attempt to re-start its associated application, job, or task.
    - Session filtering, which provides additional granularity in the display of session data.
- Trap management: NetView can receive SNMP traps, and will convert them to alerts and status updates. SNA alerts can be converted to SNMP traps and forwarded to SNMP management applications.
- AON IP resource management: An easy-to-use interface allows the display of all critical CS for OS/390 IP resources, as well as adding/deleting resources from automation policy, changing the policy for a selected resource, and displaying any outstanding timers for the resource.

Automation

- e-mail support: NetView operators can now send e-mails directly from NetView via a panel interface using SMTP.

NetView Management Console (NMC) and Graphical Management

The NetView management console (NMC) of Tivoli NetView for OS/390 graphically displays the resources that represent a network, a portion of a network, or a group of networks at various levels of detail. These views show the network and system resources that you are monitoring. The NMC is a client/server application which consists of:

- NetView Management Console server: communicates with Resource Object Data Manager (RODM) through Graphic Monitor Facility host subsystem (GMFHS) to get resource information, and provides information for display on the console. The server can be run on AIX, OS/2, and Windows NT.
- NetView Console Java client: graphically displays network information from the NMC server with colors to indicate status of resources, and provides real-time monitoring of the network.

NMC enables you to:

- Display network topology

- Specify which resources are critical to your network and place them in exception views to notify operators when those resources fail.
- Mark resources to show that they are being serviced.
- Issue commands from context menus, or use a command line to issue your own commands.
- Customize specific views.
- Display a list of events received for a selected resource.

NMC provides the built-in Java run-time environment, which eliminates the need for a separate installation and configuration of the Java Development Kit on a workstation. The productivity kit is also shipped with NMC, which provides the ability to write console-based Java applications, and to extend console operation using plug-ins.

More detailed information on Tivoli NetView for OS/390 can be found at the URL below:

```
http://www.tivoli.com/nv390/
```

1.4 OS/390 system environment at ITSO Raleigh

At ITSO Raleigh, we have one IBM 9672 on which several levels of OS/390 are running.

As a test environment in our project, we used three LPARs which OS/390 V2R8 is running, namely RA03, RA28, and RA39 respectively. All systems share the same libraries and are connected to each other using a coupling facility. For network attachment, we have two EN/TR OSA-2 adapters and one ATM OSA-2 adapter configured.

Figure 7 on page 23 shows the overview of our system environment and applications running on a specific OS/390 image that will be discussed in the following chapters.

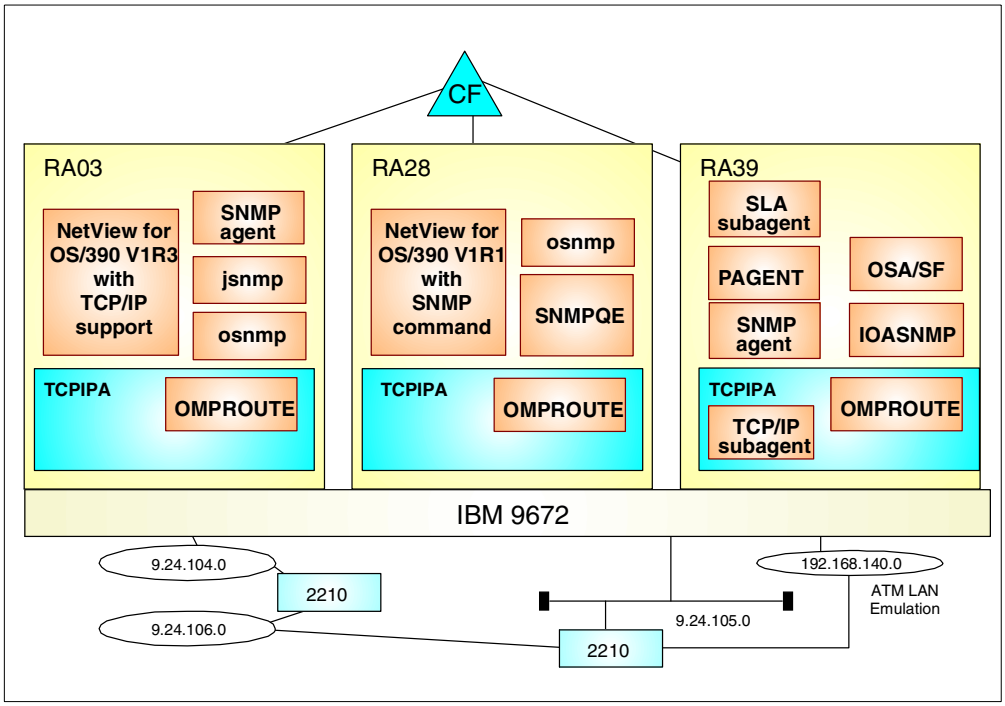


Figure 7. OS/390 system environment at ITSO Raleigh

Chapter 2. SNMP configuration in OS/390 UNIX

This chapter describes the detailed configuration information of the SNMP components shipped with CS for OS/390 V2R8 IP. The following SNMP components will be discussed in this chapter:

- SNMP agent
- UNIX SNMP manager
- TCP/IP SNMP subagent
- OMPROUTE SNMP subagent
- SLA SNMP subagent

Consult Appendix A, “Working with the NetView SNMP command” on page 165 for information on the NetView SNMP command.

2.1 SNMP implementation overview

The following section provides an overview for implementing the SNMP agent, SNMP subagents, and SNMP manager in CS for OS/390 V2R8 IP.

Before you bring up and test the SNMP agent, be sure the syslog daemon (syslogd) is started. Syslog daemon is a server process that must be started as one of the first processes in your OS/390 UNIX environment. CS for OS/390 server applications and components use syslogd for logging purposes and can also send trace information to syslogd. If syslogd is not started, application log data may appear on the MVS console. For more information about configuring the syslog daemon, please refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513.

2.1.1 SNMP agent

Configure the SNMP agent based upon your security needs. The SNMP agent for CS for OS/390 V2R8 IP accepts both SNMPv1 and SNMPv2c requests for community-based security. The SNMP agent can be configured to also use the SNMPv3 user-based security model and the view-based access control model. At ITSO Raleigh, we implemented the SNMPv3 security level with privacy.

SNMPv3 allows the agent to be configured dynamically, from either a local or remote host, and to make changes while the agent is running. The SNMPv3 framework allows you to make either temporary or premanent configuration changes depending on the storage type setting for a row in a table. By default, the dynamic configuration changes are written out in the SNMPD.CONF file so it remains in effect in case the agent is restarted. The MVS `modify` command may be used to change the time interval for which SNMPD.CONF is updated. See the *OS/390 SecureWay Communications Server IP User’s Guide*, GC31-8514 for more information.

SNMPv3 provides a more powerful and flexible framework for message security and access control. Message security involves providing:

- Data integrity checking, to ensure that the data was not altered in transit

- Data origin verification, to ensure that the request or response originates from the source from which it claims to have come
- Message timeliness checking and, optionally, data confidentiality, to protect against eavesdropping

If you want to use only community-based security (SNMPv1 and SNMPv2c) without the view-based access model, you may put your definitions in the PW.SRC and SNMPTRAP.DEST files.

Note: If the SNMPD.CONF file is found, the PW.SRC and SNMPTRAP.DEST files will not be used.

2.1.2 SNMP manager

CS for OS/390 V2R8 IP provides two different SNMP manager functions, that is the UNIX System Services shell command `osnmp` and the NetView SNMP command. We decided to use the `osnmp` command in our test because only the `osnmp` command supports the SNMPv3 operations.

The `osnmp` command is used to send SNMP requests to SNMP agents on local or remote hosts. The requests can be SNMPv1, SNMPv2 or SNMPv3. For SNMPv2 and SNMPv3 requests, the winSNMPname specified on an entry in the OSNMP.CONF file 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 SecureWay Communications Server IP User's Guide*, GC31-8514.

2.1.3 SNMP subagents

There are three SNMP subagents shipped with CS for OS/390 V2R8 IP:

- The TCP/IP subagent reports information about the TCP/IP stack. The TCP/IP subagent can also retrieve ATM OSA-2 data from the Open Systems Adapter Support Facility (OSA/SF) for support of ATM management.
- The OMPROUTE subagent reports information specific to OSPF.
- The SLA subagent reports information about defined service policies and performance statistics related to traffic using those policies.

For the TCP/IP subagent configuration, two statements are available in TCPIP.PROFILE, which are ITRACE and SACONFIG. ITRACE is used to turn on or off the internal trace for TCP/IP. This function should be used only when requested by an IBM service representative. The SACONFIG statement is used 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. The TCP/IP subagent provided by CS for OS/390 V2R8 IP will connect to OSA/SF to provide for ATM management. 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 OS/390 UNIX transport application that interconnects a subagent with OSASF1.

The sample procedures for IOAOSASF and IOASNMP can be found in IOA.AIOASAMP. These sample procedures and all entities that they call are provided with OSA/SF. OSA/SF TCP/IP subagent support is requested by specifying the OSASF parameter on the SACONFIG in TCPIP.PROFILE.

The OMPROUTE subagent implements RFC 1850 for the OSPF (Open Shortest Path First) protocol. The ROUTESA_CONFIG statement is used in the OMPROUTE configuration file to configure the OMPROUTE subagent.

The SLA subagent implements the SLA Performance Monitor MIB (SLAPM-MIB). It provides monitoring of service policies through PolicyStats, PolicyMonitor, and Subcomponent tables. The SLA subagent communicates with the SNMP agent to provide GET and SET support for various MIB tables, objects and trap generation. Traps can be generated for various performance out-of-bounds conditions, for example, maximum delay exceeded, and for significant events such as policy deletion. The SLA Performance Monitor subagent runs as an independent UNIX process in a separate MVS address space.

See Figure 8 for an overview diagram of the CS for OS/390 V2R8 IP SNMP support:

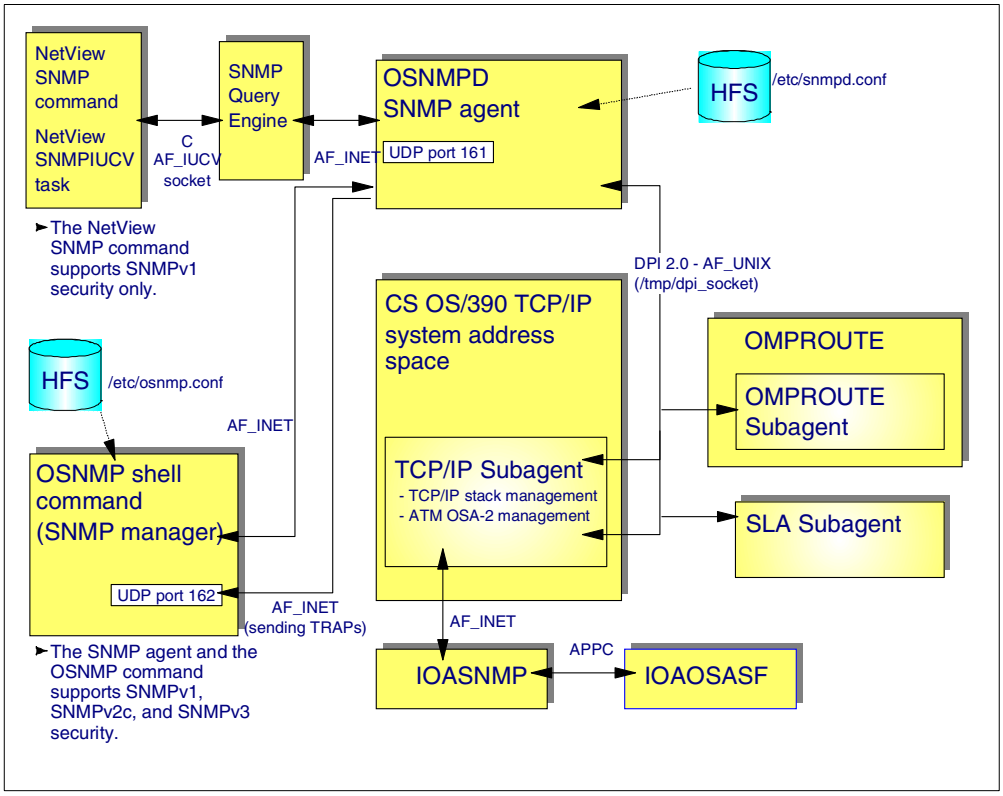


Figure 8. SNMP support in CS for OS/390 V2R8

2.2 Configure the SNMP agent

The following section details the configuration steps necessary to initialize the SNMP agents and subagents within CS for OS/390 V2R8 IP.

Configuration for the SNMP agent depends on which types of security will be used. If only community-based security will be used, the legacy PW.SRC and SNMPTRAP.DEST files can be used. However, if USM security is desired, that is if the SNMPv3 protocol is used, the SNMPD.CONF file is needed.

The required configuration files vary with the security level you are going to implement. See Table 7 for information about which files are required for which security level.

Table 7. Configuration files required depending on the SNMP security level

Configuration file	Community-based security	USM
PW.SRC	Yes	No
SNMPTRAP.DEST	Yes	No
OSNMPD.DATA	Yes	Yes
SNMPD.CONF	No	Yes
SNMPD.BOOTS	No	Yes
Note: The USM configuration also supports community-based security, which is supported by both SNMPv1 and SNMpv2c security models.		

The PW.SRC and SNMPTRAP.DEST files are used for the legacy configuration and will not be discussed in this redbook. Please refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513 for detailed information on these files.

To configure the SNMP agent with USM, follow the steps below:

1. Configure TCPIP.PROFILE.
2. Manually configure the SNMPD.BOOTS file, if needed.
3. Specify MIB objects and their values in the OSNMPD.DATA file.
4. Create authentication and (optionally) privacy keys, if you are using the SNMPv3 security level.
5. Configure the SNMP agent for security and trap destinations.

2.2.1 TCPIP.PROFILE

For the SNMP agent we reserved port 161, and for the osnmp command ports 162 and 16200 to receive the trap information. The port reservation is configured in TCPIP.PROFILE:

```
; Member TCPIP.TCPPARMS(PROF39A)
;*********************************************************************
PORT
   161 UDP OSNMPD              ; SNMP Agent
   162 UDP OMVS                ; osnmp command (SNMP Manager)
 16200 UDP OMVS                ; osnmp command (SNMP Manager)
```

If the agent is started from the OS/390 shell, reserve port 161 for OS/390 UNIX by typing OMVS instead of the MVS procedure name. If you use NetView as a manager you should reserve port 162 for the SNMP Query engine or leave it unreserved.

To configure SNMP subagents shipped with CS for OS/390, you have to update the following configuration files depending on the subagent used:

- TCP/IP subagent without SNMP ATM management

  Update the SACONFIG and ITRACE statements in PROFILE.TCPIP.

- TCP/IP subagent with SNMP ATM management

  Customize the OSA/SF configuration, the IOASNMP address space, and the DEVICE / LINK statement in PROFILE.TCPIP if necessary, in addition to the configuration of the SACONFIG and ITRACE statements in PROFILE.TCPIP.

- OMPROUTE subagent

  Update the ROUTESA_CONFIG in the OMPROUTE configuration.

For more information on the SNMP subagent configuration, consult 2.3, “Configure the SNMP subagent on OS/390” on page 38.

2.2.2 SNMPD.BOOTS

The SNMP agent uses the SNMPD.BOOTS configuration file to support SNMPv3 security. This file contains agent information used to authenticate the SNMPv3 requests. The SNMPD.BOOTS file keeps the engine ID of the SNMP agent and the number of times the agent reboots. If no SNMPD.BOOTS file exists when the agent is started, the agent creates one. If a file does exist, the agent uses the values specified in the file for setting its engine ID and engineBoots values. If the file exists but contains invalid values for engine ID or engineBoots, the agent issues a message and terminates. This file contains two parameters:

- Engine ID, which uniquely identifies, in an administrative domain, the SNMP agent.
- engineBoots, which is a count of the number of times the SNMP agent has rebooted/re initialized since the last configuration was made.

Engine ID contained in SNMPD.BOOTS is interpreted as follows:

```
00000002 00000000xxxxxxxx
```

The first four octets are set to the private enterprise number for IBM, and the value is 2. The remaining eight octets are specified in an enterprise specific way. SNMP agent in OS/390 V2R8 IP uses its IP address, which is given by gethostname API, for this field.

Localized keys used for SNMPv3 security are derived using the engine ID or a local IP address if you use an IP address for your agent’s engine ID.

If you want to use a VIPA address for an engine ID, ensure both secret keys and the engine ID are derived from the VIPA address. At the initialization phase, if the /etc/snmpd.boots file does not exist, the SNMP agent creates its engine ID using an IP address associated with one of the physical interfaces configured for the TCP/IP stack that it is running. It ignores a VIPA address even if it is configured as a primary IP address on a TCP/IP stack. Therefore you will have to update the

/etc/osnmpd.boots file manually for the engine ID to be created from the VIPA address.

You can just as easily use an arbitrary engine ID, such as 1234. As long as you use that engine ID when creating localized keys for use at that SNMP agent, that engine ID will work just as well as the one the SNMP agent creates. You should take care, however, that there are no duplicate engine IDs within the administrative domain.

The search order of SNMPD.BOOTS information is as follows:

1. The name of an HFS file or an MVS file specified by the SNMPD_BOOTS environment variable
2. /etc/snmpd.boots

We configured /etc/snmpd.boots manually to use a physical IP address (9.24.104.149) as follows:

```
000000020000000009186895 0000000001
```

2.2.3 OSNMPD.DATA

OSNMPD.DATA contains values for selected MIB objects. You can customize those values for your environment. If no OSNMPD.DATA file is found, the defaults for these MIB objects are as follows:

Table 8. OSNMP.DATA: MIBs' default values

Object Name	Default Value
sysDescr	The value of HOSTNAME environment variable.
sysContact	SNMPBASE-Unspecified
sysLocation	SNMPBASE-Unspecified
sysName	SNMPBASE-Unspecified
sysObjectId	1.3.6.1.4.1.2.3.13
sysServices	A single octet that defaults to 0
snmpEnableAuthenTraps	Default value is 2, which means authentication traps are disabled
saDefaultTimeout	5 seconds

The first file found in the following search order will be used to obtain the OSNMPD.DATA information:

1. The name of an HFS file or MVS file specified by the OSNMPD_DATA environment variable
2. /etc/osnmpd.data HFS file
3. The data set specified on the OSNMPD DD statement in the agent procedure
4. jobname.OSNMPD.DATA, where jobname is the name of the job used to start the SNMP agent
5. SYS1.TCPPARMS(OSNMPD)

6. hlq.OSNMPD.DATA, where hlq either defaults to TCP/IP or is specified on the DATASETPREFIX statement in the TCPIP.DATA file being used

OSNMPD picks up the HOSTNAME from TCPDATA and translates this name using a domain name server into its agent address. We used the following OSNMPD.DATA definitions:

```
#
# osnmpd.data (/etc/osnmpd.data)
#
# Sample file for setting MIB variables and options for
# the SNMPv3 Agent provided by OS/390
#
# Licensed Materials - Property of IBM
# "Restricted Materials of IBM"
# 5647-A01
# (C) Copyright IBM Corp. 1996, 1999
# Status = CSV2R8
#
# Flag Reason   Release  Date   Origin    Description
# ---- -------- -------- ------ --------  -----------------------
# $A1= PQ22119  TCPB340  990114 SHAGGAR : Enable multiple DPI v1
#                                        subagents
#
  sysDescr "SNMPv3 agent version 1.0 with DPI version 2.0"
  sysContact  "Tatsuhiko Kakimoto"
  sysLocation "ITSO Raleigh, IBM"
  sysName "CS/390 for OS/390 V2R8 IP on SYSTEM39"
  sysObjectID     "1.3.6.1.4.1.2.3.13"
  snmpEnableAuthenTraps 1
  saDefaultTimeout    6
  saMaxTimeout  700
# saAllowDuplicateIDs must be set to 1 to allow multiple DPI version 1
#  subagents
  saAllowDuplicateIDs   1
  dpiPathNameForUnixStream "/tmp/dpi_socket"
# Default value of sysServices indicates support for
#  internet, end-to-end, and application layers as
#  defined in RFC 1907.
  sysServices       76
```

The default value of sysObjectID is equivalent to ibmTcpIpMvs in the ibmAgents subtree; this is the sysObjectID representing SecureWay Communications Server on OS/390.

2.2.4 Key generation for authentication and privacy

RFC 2574 describes the User-based Security Model (USM) for message security. USM uses a basic concept of a user, on whose behalf SNMP messages are generated. For USM to work, the user must be defined to both the manager and the agent. For an authenticated request on behalf of the user, both the manager and the agent must know a set of one or more keys to be used in processing the message. These keys are the authentication key and optionally, the privacy key. Authentication is required to process SNMPv3 requests unless the requested security level is noAuth.

The authentication protocols currently supported with USM are:

- HMAC-Message Digest 5, or HMAC-MD5, which uses a 128-bit key (16 bytes)
- HMAC-Secure Hash Algorithm, or HMAC-SHA, which uses a 160-bit key (20 bytes)

The privacy protocol supported with USM, if the separate product is applied, is Cipher Block Chaining Data Encryption Standard (DES), or CBC 56-bit DES. All of the protocols are based on symmetric cryptography (private key) mechanisms.

Encryption is an optional feature of USM. It will not be in the base product. However, it will be available as a separately ordered product where export laws allow. See Table 4 on page 17 for FMIDs of encryption features in CS for OS/390.

The authentication process consists of:

- Verification of the integrity of a received message
- Verification of the user on whose behalf the message was generated

An SNMP engine must have knowledge of that user, so a match is done between the userName and authKey specified in the osnmp.conf file for an SNMP manager and the snmpd.conf file for an agent.

CS for OS/390 IP provides a facility called pwtokey, to convert passwords into authentication keys. It takes as input the password and an identifier of the agent (the engine ID or the IP address).

The output generates two kinds of keys: one that is localized (usable only in the specified agent) and one that is not localized. It is recommended that you use localized keys in an SNMP agent configuration and non-localized keys in an SNMP manager configuration, that is, in /etc/snmpd.conf and /etc/osnmp.conf respectively in most installations.

At ITSO in Raleigh, we used the 16-byte HMAC-MD5 protocol for USM authentication and privacy. These are samples of the pwtokey output obtained in our implementation.

```
                                  1              2         3
MARTIN @ RA39:/u/martin>pwtokey -p HMAC-MD5 -u auth authpass 9.24.104.149
Display of 16 byte HMAC-MD5 authKey:
  eb81360f73adf8d6967eaadec03dff45     4

Display of 16 byte HMAC-MD5 localized authKey:
  b186f7ab80c9e7f8cda3dbc6f6203f97     5

                                  1              2         3
MARTIN @ RA39:/u/martin>pwtokey -p HMAC-MD5 -u priv privpass 9.24.104.149
Display of 16 byte HMAC-MD5 privKey:
  7865f0c012464308a5a4deead83903c3     4

Display of 16 byte HMAC-MD5 localized privKey
  61a5dfe8a7277f1f982edce11e9fc633     5
```

Figure 9. Pwtokey utility to generate authentication and privacy keys

1 Indicates we want to generate keys with the HMAC-MD5 protocol.

2 Indicates the passwords we use in our implementation. If you use the same password for both, the output for the privacy key will be the same as the output for

the authentication key, except that the localized privacy key for 20-byte HMAC-SHA will be truncated to 16 bytes. CBC DES requires the use of 16-byte (32-hex digits) keys for privacy. At ITSO, we use different passwords for authentication and privacy.

3 Is the agent IP address that was used to create the engine ID. If you choose to use your own algorithm for creating an engine ID, you have to invoke the pwtokey command with the -e flag and the engine ID instead of the IP address.

4 Is the non-localized key used in the osnmp.conf file for the SNMP manager.

5 Is the localized key used in the snmpd.conf file for the SNMP agent.

Note: In our implementation we use privacy keys for encryption purposes. Encryption is supported only as a separately ordered feature on the base OS/390 product.

2.2.4.1 Create security keys from an engine ID

If you want to use a customized engine ID that does not have any relationship with IP addresses configured on a TCP/IP stack, secret keys have to be created from the engine ID. In this case, issue the `pwtokey` utility with the `-e` option.

```
KAKKY @ RA03:/u/kakky>cat /etc/snmpd.boots
 03 0000000005            1

                                2
KAKKY @ RA03:/u/kakky>pwtokey -e -p HMAC-MD5 -u auth authpass 03
Display of 16 byte HMAC-MD5 authKey:
  eb81360f73adf8d6967eaadec03dff45     3

Display of 16 byte HMAC-MD5 localized authKey:
  b3c1483c0bc1945a9f08914123cbe935

KAKKY @ RA03:/u/kakky>pwtokey -e -p HMAC-MD5 -u priv privpass 03
Display of 16 byte HMAC-MD5 privKey:
  7865f0c012464308a5a4deead83903c3     3

Display of 16 byte HMAC-MD5 localized privKey:
  0944ee5cd0bd8e1d748bada4c682a732
```

In this sample implementation, an engine ID has been configured using an arbitrary value, that is 03 1. The security keys are created by issuing the `pwtokey` command with `-e` option 2 and the engine ID configured 3.

Regardless of which types of engine ID you use, ensure secret keys have been derived from the value that has been used to configure the agent's engine ID. This can be either a customized engine ID or an IP address that has been used to create the engine ID.

2.2.5 Configure the SNMP agent for security and trap destinations

The SNMPD.CONF file defines the SNMP agent security and notification destinations. (A trap is one type of notification. The other type, informs, are not supported in CS for OS/390.) If the SNMPD.CONF file exists, the agent can support SNMPv1, SNMPv2c, and SNMPv3 requests. If no SNMPD.CONF file exists, the agent will support only SNMPv1 and SNMPv2c requests.

To access SNMPD.CONF information, the search order is:

1. An HFS file or an MVS file specified by the SNMPD_CONF environment variable
2. /etc/snmpd.conf

We configured the SNMP agent to support SNMPv3, SNMPv2c and SNMPv1 requests, so we put our definitions in the /etc/snmpd.conf file.

SNMPv3 allows you to configure the agent dynamically, from either a local or remote host, and to make changes while the agent is running. The dynamic configuration changes may be written out in the snmpd.conf file so it remains in effect in case the agent is restarted.

If you want to use only community-based security (SNMPv1 and SNMPv2c) without the view-based access model, you may put your definitions in the PW.SRC and SNMPTRAP.DEST files. You can still use the SNMPD.CONF file for just SNMPv1 and SNMPv2c security. One of the advantages of using the SNMPD.CONF configuration file, while you are using community-based security only, is to make use of the VACM access control with SNMPv1 and SNMPv2. Using PW.SRC does not provide this level of access control.

Note: If the SNMPD.CONF file is found, the PW.SRC and SNMPTRAP.DEST files will not be used.

The following nine types of entries could be configured in the SNMPD.CONF file.

Table 9. The SNMPD.CONF entries

Entry Name	Description
USM_USER	Defines a user for the User-Based Security Model (USM).
VACM_GROUP	Defines a security group (made up of users or communities) for the View-Based Access Control Model (VACM).
VACM_VIEW	Defines a particular set of MIB objects, called a view, for the View-Based Access Control Model.
VACM_ACCESS	Identifies the access permitted to different security groups for the View-Based Access Control Model.
NOTIFY	Identifies management targets to receive notifications.
TARGET_ADDRESS	Defines a management application's address and identifies parameters to be used in sending notifications.
TARGET_PARAMETERS	Defines the message processing and identifies security parameters to be used in sending notifications to a particular management target.
COMMUNITY	Defines a community for community-based security.
DEFAULT_SECURITY	Identifies the default security posture to be configured for the SNMP agent.

The following configuration file was used to configure the SNMP agent in our environment on RA39:

```
#
# /etc/snmpd.conf : CS for OS/390 V2R8 IP on SYSTEM 39
#
# Licensed Materials - Property of IBM
# "Restricted Materials of IBM"
# 5647-A01
# (C) Copyright IBM Corp. 1997, 1998
#
#-------------------------------------------------------------------------------------------------------
# Notes
# - All values for an entry must be on the same line.
# - All keys need to be regenerated using the pwtokey command in order
#   for these sample entries to actually be used.
# - In this sample:
#   - Keys are generated for use with engineID 000000020000000009186895
#   - Authentication keys were generated with password of
#     "authpass" and are localized keys.
#     Note: The password could be different for every user              owed.
#   - Privacy keys generated with a password of "privpass".
#   - Entries defined to use encryption support, which is available only
#     as a separately orderable feature on the base OS/390 product, are
#     included below but commented out.
#-------------------------------------------------------------------------------------------------------

#-------------------------------------------------------------------------------------------------------
# USM_USER entries  1
# Format is:                                                                    11
#  userName engineID authProto authKey privProto privKey keyType storageType
#
#-------------------------------------------------------------------------------------------------------
USM_USER u03  - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  DES 61a5dfe8a7277f1f982edce11e9fc633 L -
USM_USER  u03n - none -   -   -  L -            2
USM_USER  u28 - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  DES 61a5dfe8a7277f1f982edce11e9fc633 L -
USM_USER  u28n - none -   -   -  L -
USM_USER  u39 - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  -   -  L -

#-------------------------------------------------------------------------------------------------------
# VACM_GROUP entries        3
# Format is:                                  11
#  groupName securityModel securityName storageType
#-------------------------------------------------------------------------------------------------------
VACM_GROUP group1 USM     u03  -
VACM_GROUP group1 USM     u03n -
VACM_GROUP group1 USM     u28  -     12
VACM_GROUP group1 USM     u28n -
VACM_GROUP group1 USM     u39  -

#-------------------------------------------------------------------------------------------------------
VACM_GROUP group3 SNMPv1  publicv1  -
VACM_GROUP group3 SNMPv2c publicv2c -         13

VACM_GROUP group4 SNMPv1  MVSsubagent  -
VACM_GROUP group4 SNMPv2c MVSsubagent  -         14

#-------------------------------------------------------------------------------------------------------
# VACM_VIEW entries   4
# Format is:                                  11
#  viewName viewSubtree viewMask viewType storageType
#-------------------------------------------------------------------------------------------------------
VACM_VIEW bigView        internet   - included -

VACM_VIEW prettyBigView  internet   - included -
VACM_VIEW prettyBigView  interfaces - excluded -

VACM_VIEW mediumView     system     - included -
VACM_VIEW mediumView     interfaces - included -
VACM_VIEW mediumView     tcp        - included -
VACM_VIEW mediumView     udp        - included -
VACM_VIEW mediumView     icmp       - included -

VACM_VIEW smallView      snmp       - included -

VACM_VIEW subagentView   dpiPort    - included -  17

#-------------------------------------------------------------------------------------------------------
# VACM_ACCESS entries    5
# Format is:                                                                                  11
#            group context context security security  read     write        notify  storage
#             Name  Prefix  Match   Level    Model    View     View         View     Type
```

```
#-------------------------------------------------------------------------------------------------
VACM_ACCESS  group1  -      -      AuthPriv   USM     bigView   prettyBigView bigView   -
VACM_ACCESS  group1  -      -      AuthNoPriv USM     bigView   prettyBigView bigView 15-
VACM_ACCESS  group1  -      -    noAuthNoPriv USM     bigView   smallView     bigView 15-

VACM_ACCESS  group3  -      -    noAuthNoPriv SNMPv1  bigView   bigView       bigView 16-
VACM_ACCESS  group3  -      -    noAuthNoPriv SNMPv2c bigView   bigView       bigView 16-

VACM_ACCESS  group4  -      -    noAuthNoPriv SNMPv1  subagentView   -            -    - 17
VACM_ACCESS  group4  -      -    noAuthNoPriv SNMPv2c subagentView   -            -    - 17

#-------------------------------------------------------------------------------------------------
# NOTIFY entries  6
# Format is:                11
#  notifyName tag type storageType
#-------------------------------------------------------------------------------------------------
NOTIFY notify1 traptag trap -

#-------------------------------------------------------------------------------------------------
# TARGET_ADDRESS   7
# Format is:                                                                        11
#           target  tDomain tAddress          tagList  target  timeout retry  storage
#           AddrName                                   Params         Count   Type
#-------------------------------------------------------------------------------------------------
                21                 18
TARGET_ADDRESS Target1 UDP 9.24.104.113:16200 traptag trapparms1  -   -        -
TARGET_ADDRESS Target2 UDP 9.24.104.113:16200  traptag trapparms2  -   -       -
TARGET_ADDRESS Target3 UDP 9.24.104.113:16200  traptag trapparms3  -   -       -
                                  19
TARGET_ADDRESS Target6 UDP 127.0.0.1:16200     traptag trapparms5  -   -       -
                                  20
TARGET_ADDRESS Target8 UDP 9.24.104.42:162     traptag trapparms1  -   -       -
#-------------------------------------------------------------------------------------------------
# TARGET_PARAMETERS   8
# Format is:                                                          11
#                 params    mpModel security security   security   storage
#                  Name              Model    Name       Level      Type
#-------------------------------------------------------------------------------------------------
TARGET_PARAMETERS trapparms1 SNMPv1  SNMPv1  publicv1  noAuthNoPriv  -
TARGET_PARAMETERS trapparms2 SNMPv2c SNMPv2c publicv2c noAuthNoPriv  -
TARGET_PARAMETERS trapparms3 SNMPv3  USM     u03       AuthNoPriv    -
TARGET_PARAMETERS trapparms4 SNMPv3  USM     u28       AuthPriv    -
TARGET_PARAMETERS trapparms5 SNMPv3  USM     u39       AuthNoPriv    -

#-------------------------------------------------------------------------------------------------
# COMMUNITY  9
# Format is:                                                                  11
#        community    security    security       netAddr    netMask  storageType
#         Name         Name         Level
#-------------------------------------------------------------------------------------------------
COMMUNITY publicv1    publicv1     noAuthNoPriv  0.0.0.0     0.0.0.0      -
COMMUNITY publicv2c   publicv2c    noAuthNoPriv  0.0.0.0     0.0.0.0      -
COMMUNITY MVSsubagent MVSsubagent  noAuthNoPriv  0.0.0.0     0.0.0.0      -
#-------------------------------------------------------------------------------------------------

## DEFAULT_SECURITY  10
# Format is:
#  securityPosture password privacy
#-------------------------------------------------------------------------------------------------
DEFAULT_SECURITY semi-secure defaultpassword no

#-------------------------------------------------------------------------------------------------
# Any SNMP agent configuration entries added by dynamic configuration
# (SET) requests get added to the end of the SNMPD.CONF file.
#-------------------------------------------------------------------------------------------------
```

Figure 10. SNMP agent configuration file on RA39

1 This section describes the users for the User-based Security Model (USM). The name must be unique in the agent configuration and it is associated with a key that is used in the authentication process following the protocol specified. A localized key created by the pwtokey utility is recommended to be used for this entry. When you use a non-localized key, you have to specify *N* for the keyType keyword.

2 This shows the usage of USM security. We configure three different levels of the USM security: authentication and privacy, authentication only, and no security.

3 This section associates the user name previously defined (or generically, the SNMPv1 and SNMPv2c community) with a group name. We used three groups, one is for the SNMPv3 security domain and all members belong to group1 (12). For the SNMPv1 and SNMPv2 community domain, group3 is configured (13), and each SNMP subagent will be a member of group4 (14). The community name for SNMP subagents, MVSsubagent in this case, has to match the value in the subagent configurations.

4 This section defines the view name associated with the MIB object prefix of the MIB tree in the view. Valid values are both textual object ID (OID) (for example, internet) or numerical (for example, 1.3.6.1.2.1...).

5 This section associates the group name with the views previously defined, stating the security level and the authentication protocol to be used. For example, a user belonging to group1 with AuthNoPriv security level can read all MIB internet information but cannot set (write) the interface values.

For the SNMP subagents that are members of group4, we allow them to read the view, *subagentView*, which has been defined in the VACM_VIEW statement (17).

6 This defines the notification information. The only type of notification supported is trap.

7 This section defines the notification target(s). It should be an SNMP manager (for example, Tivoli NetView) which will receive the generated traps (usually on port 162) to take the appropriate actions. In our environment, the SNMP agent will send traps to the `osnmp` command running over UDP port number 16200 on RA03 (IP address 9.24.104.113 18) and local host (RA28, IP address 127.0.0.1 19), and the NetView SNMP command on RA28 (IP address 9.24.104.42 20).

8 This defines the security and the message processing model to be used in sending the notifications to the previously defined targets.

The index of the TARGET_ADDRESS statement is the targetAddrName (21) for the traps. If you need to send traps to one more different addresses, then the targetAddrName for each destination should be different.

For traps to be sent, the user should be granted access to the part of the subtree that is required to send out traps. In our sample configuration, the TARGET_PARAMETERS statement

```
TARGET_PARAMETERS trapparms3 SNMPv3  USM     u03  AuthNoPriv -
```

specifies that the trap should be sent on behalf of user u03. Because user u03 belongs to group1, in order to receive traps, group1 has to have access to bigView for notifyView (15). Note that smallView can access only the SNMP group and the traps are not forwarded.

Therefore, the VACM_ACCESS statement was configured as follows:

```
VACM_ACCESS  group1 - - AuthNoPriv  USM  bigView  prettyBigView bigView -
```

This granted the access required to send the traps.

Similarly, for traps in the SNMPv1 and SNMPv2 domains, the groups have to have access to bigView to notify View in our configuration (16).

9 This section describes the community security environment for requests coming from SNMP managers that do not support (or do not want to use) the SNMPv3 security model.

10 This defines the security default values to be applied to the previous entries. The default security is intended as a way for customers to create an easy initial configuration. It defines a user, a group, some views, and some access rights. These default definitions may be used to try out SNMPv3, but after that, you would probably define them more explicitly.

11 This parameter indicates the type of storage in which the definitions are to be maintained. The default taken is nonvolatile which means the information will persist across reboots of the agent, but it can be changed by dynamic configuration requests.

For more information about syntax and definitions, please refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513, and to *OS/390 SecureWay Communications Server IP User's Guide*, GC31-8514.

2.3 Configure the SNMP subagent on OS/390

There are three SNMP subagents shipped with CS for OS/390 V2R8 IP:

- The TCP/IP subagent reports information about the TCP/IP stack. The TCP/IP subagent will also connect to OSA/SF to provide for ATM management.
- The OMPROUTE subagent reports information specific to OSPF.
- The SLA subagent reports information about defined service policies and performance statistics related to traffic using those policies.

At ITSO in Raleigh, we have implemented the three subagents including OSA/SF ATM for the TCP/IP subagent.

2.3.1 TCP/IP subagent configuration

The TCP/IP subagent in CS for OS/390 is an OS/390 UNIX application that runs in its own task in the TCP/IP address space. This subagent supports many standard (RFC-based) MIB objects in addition to enterprise-specific MIB objects including:

- The IBM 3172 enterprise-specific MIB.
- A TCP/IP for Multiple Virtual Storage (MVS) enterprise-specific MIB. It defines objects to extend standard MIB tables, supports retrieval and change of TCP/IP address space configuration parameters, and provides management support for the environments where Asynchronous Transfer Mode (ATM) is used.

There are two statements in the PROFILE.TCPIP used to configure the TCP/IP subagent, the SACONFIG and ITRACE statements. The ITRACE statement is used to determine what trace information, if any, should be recorded by the subagent. The parameter for configuring the TCP/IP subagent is the SACONFIG statement in the PROFILE.TCPIP. 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.

We used the following SACONFIG definition on RA39 at ITSO:

```
; ---------------------------------------------
SACONFIG  COMMUNITY MVSsubagent 1
 ENABLED
 AGENT 161      2
 OSASF 760      3
 ATMENABLED     4
 SETSENABLED    5
```

1 This specifies the community name which the subagent will use in establishing contact with the SNMP agent. For the TCP/IP SNMP subagent to communicate with the CS for OS/390 V2R8 IP SNMP, the community name specified (or defaulted) on the COMMUNITY keyword must match the one that is defined in the PW.SRC or SNMPD.CONF data set used by the SNMP agent or specified (or defaulted) on the -c parameter when the SNMP agent is started. SNMP subagents use the SNMPv1 protocol with this community name to establish communication with SNMP agents.

2 There are two primary 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 port 161.

3 This parameter specifies which port IOASNMP should use to listen for connections from subagents to OSA/SF. It is recommended that the OSASF port be reserved by also specifying it on the PORT statement.

4 ATM Management support must be enabled by specifying ATMENABLED. Omission of ATMENABLED when TCP/IP is started will result in no ATM Management support.

5 We enabled the SET command at the TCP/IP subagent. You will see the complete lists of MIBs that CS for OS/390 V2R8 IP supports in Appendix C, “Management Information Base (MIB) objects” on page 213. It also shows the access permission for the objects such as read-only and read-write.

If SACONFIG is not defined, the following default values are set:

- The TCP/IP subagent is started at TCP/IP initialization.
- The default value for the community name is public.
- Default port 161 to communicate with the SNMP agent.
- The SNMP ATM management is not enabled.

2.3.2 TCP/IP subagent with the SNMP ATM management

The TCP/IP subagent can retrieve ATM OSA-2 data from the Open Systems Adapter Support Facility (OSA/SF) for support of ATM management. The ATM management data is organized into MIB tables defined in the IBM MVS TCP/IP Enterprise Specific MIB. The ATM management data is available for any ATM OSA-2 adapter cards, whether or not they are being used by the TCP/IP instance to which they are defined. As long as they are defined by DEVICE and LINK profile statements, SNMP can retrieve ATM management data for them even if they are defined to VTAM, a different TCP/IP instance, or if they are configured for ATM LAN emulation mode and are defined to this TCP/IP instance or another TCP/IP instance in the same MVS image as a LAN channel station (LCS) device.

The primary purpose of OSA/SF is to manage OSA adapters. It has been extended to support ATM management using SNMP. The following instances must be started on every MVS image where ATM management support is needed:

- IOAOSASF, which is the main OSA/SF address space
- IOASNMP, which is the OSA/SF-provided OS/390 UNIX transport application that interconnects one or more TCP/IP subagent with an OSA/SF address space
- TCP/IP stack
- TCP/IP subagent, which must be configured with the ATMENABLED keyword on the SACONFIG statement in TCPIP.PROFILE
- SNMP agent

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 OS/390 UNIX 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.
4. Start the SNMP agent, OSNMPD, for each TCP/IP instance where ATM management support is desired.

For a description of the ATM OSA-2 adapter management, see *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513 and *Planning for the S/390 Open Systems Adapter (OSA-1, OSA-2) Feature*, GC23-3870.

2.3.2.1 Configure OSA/SF address space

At ITSO, we have copied and renamed the sample IOAOSASF to OSASF. The sample started procedure can be found as IOA.AIOASAMP(IOAOSASF).

```
//*
//*   START OSA SUPPORT FACILITY
//*
//OSASF   EXEC PGM=IOAMAIN,TIME=1440,REGION=6M,DYNAMNBR=5
//STEPLIB  DD  DSN=CEE.SCEERUN,DISP=SHR
//         DD  DISP=SHR,DSN=SYS1.SIOALMOD
//IOALIB   DD  DISP=SHR,DSN=SYS1.SIOALMOD
//IOAPROF  DD  DSN=OSASF.RA39.STARTUP.PROFILE,DISP=SHR  1
//SYSPRINT DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=121)
//SYSUDUMP DD  SYSOUT=*
```

1 In the MVS data set pointed to by IOAPROF DD we code the following:

```
SET ALIAS SYSNAME RA39
SET ALIAS CECNAME RA9672A
*
SET NAME IOACFG
```

```
    DATASET 'OSASF.&CECNAME.OSAS.CONFIG'
    UNIT  SYSALLDA
*
SET NAME IOAINX
    DATASET 'OSASF.&CECNAME.MASTER.INDEX'
    UNIT  SYSALLDA
*
SET NAME IOAMSG
    DATASET 'OSASF.&SYSNAME.MESSAGE.LOG'
    UNIT  SYSALLDA
*
SET NAME IOADSN
    DATASET 'OSASF.&CECNAME.OSASF'
    UNIT  SYSALLDA
```

2.3.2.2 Configure IOASNMP

The sample procedure for IOASNMP can be found in IOA.AIOASAMP(IOASNMP). The procedure for IOASNMP we used is:

```
//*
//* START OSA SUPPORT FACILITY SNMP COMMUNICATION FACILITY
//*
//IOASNMP EXEC PGM=IOANMAIN,TIME=1440,REGION=0K,DYNAMNBR=5
//STEPLIB  DD  DSN=SYS1.SIOALMOD,DISP=SHR
//SYSPRINT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//SYSTCPD  DD  DISP=SHR,DSN=TCPIP.TCPPARMS(TCPD&SYSCLONE.A)  2
```

2 IOASNMP supports the SYSTCPD DD statement to allocate the TCPIP.DATA data set. The environment variable RESOLVER_CONFIG also can be used for this purpose. The TCPIPJOBNAME statement in TCPIP.DATA tells IOASNMP the TCP/IP address space name with which it should establish the affinity.

We have the following statements configured in TCPIP.DATA:

```
;*********************************************************************
;                                                                    *
;   Name of Data Set:     TCPIP.TCPPARMS(TCPD39A)                    *
;                                                                    *
;*********************************************************************
TCPIPJOBNAME TCPIPA          3
HOSTNAME  MVS39A
DOMAINORIGIN  itso.ral.ibm.com
NSINTERADDR  9.24.106.15
NSPORTADDR 53
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
DATASETPREFIX TCPIP
MESSAGECASE MIXED
```

3 This is the TCPIP started procedure name on which IOASNMP should run.

For the OSA ATM subagent, the first components to start are IOAOSASF followed by the TCP/IP stack. When you start IOASNMP next you will see the following in the job log:

```
EZACDMGA Passed port
IOAC141I 15:29:36 API APPC component started
IOAC102I OSA Support Facility initialization completed 02/25/2000 15:29:36
```

Figure 11. The job log messages from IOASNMP

2.3.2.3 PROFILE.TCPIP customization

Since we used an ATM OSA-2 adapter to attach to ATM emulated LANs, ATM interfaces have to be configured as LCS devices. Up to two LAN emulation client (LEC) ports can be configured on a single ATM OSA-2 adapter. The two LEC ports on the same ATM OSA-2 cannot provide services to the same ELAN at the same time. Each LEC port can provide LEC services on either an Ethernet or a token-ring emulated LAN, but not on both ELANs concurrently.

Each LEC port can provide LEC services on either an Ethernet or a token-ring emulated LAN, but not on both ELANs concurrently.

We have configured one token-ring LEC and one Ethernet LEC on our ATM OSA-2 adapter. Each LEC joins a different emulated LAN.

In the TCPIP.PROFILE PORT statement, we have the following statements:

```
; -------------------------------------
SACONFIG  COMMUNITY MVSsubagent
 ENABLED
 AGENT 161
 SETSENABLED
 ATMENABLED                4
 OSASF 760                 5

ITRACE OFF
; ITRACE ON SUBAGENT 3      6

AUTOLOG 1
    OMPROUTA             ; OSPF   Server
    IOASNMP              ; SNMP ATM OSA-2 adapter management  7
;   OSNMPD                                                    8
ENDAUTOLOG

; -------------------------
PORT
   161 UDP OSNMPD              ; SNMP Agent
   162 UDP OMVS                ; osnmp command (SNMP Manager)
 16200 UDP OMVS                ; osnmp command (SNMP Manager)
         :
   760 TCP IOASNMP             ; IOASNMP                 9

; ******************************************
; ATM OSA Definition for SNMP management
; ******************************************
  DEVICE OSARA39A ATM   PORTNAME PORTE0            a
  LINK   OSARA39A ATM   OSARA39A

; ******************************************
; ATM OSA Definition LE
```

```
; *******************************************
  DEVICE OSA20A0 LCS   20A0                          b
  LINK OSAT20A0  IBMTR 0 OSA20A0

  DEVICE OSA20A2 LCS   20A2                          c
  LINK OSAT20A2  ETHERor802.3 1 OSA20A2

HOME                                                 d
    172.16.232.39 VIPA39A     ; VIPA
    192.168.140.12 OSAT20A0   ; ATM LANE Token-ring
    192.168.140.75 OSAT20A2   ; ATM LANE Ethernet

START OSA20A0
START OSA20A2
```

Figure 12. TCPIP.PROFILE configuration for SNMP ATM management

4 To enable SNMP ATM management, the ATMENABLED keyword for the SACONFIG statement in PROFILE.TCPIP has to be configured. The SACONFIG statement controls the operation of the subagent that runs in a TCP/IP address space as a separate task.

5 The OSA/SF parameter specifies which port IOASNMP should use to listen for connections from TCP/IP subagents. This parameter also has to be defined, because there is no default value.

6 By customizing the ITRACE statement, you can turn on the internal trace for a TCP/IP subagent. The supported levels for the TCP/IP subagent are from one to four. Trace level four gives you the most detailed information.

For the AUTOLOG statement, we specified IOASNMP 7, which is the started procedure name for IOASNMP, but not OSNMPD 8, the started procedure for the SNMP agent. This is because IOASNMP has to be started before the SNMP agent starts.

9 TCP port 760 has been reserved for IOASNMP. The same port number has to be specified for the OSASF parameter for the SACONFIG statement.

a The primary purpose of the DEVICE and LINK ATM statements is to configure ATM OSA-2 adapters in ATM native mode. However, the presence of these statements enables SNMP network management also. Therefore even if an ATM LAN emulation mode is being used instead of ATM native mode, you must configure the DEVICE and LINK ATM statements to retrieve SNMP network management information for the ATM device.

The port name specified in the DEVICE ATM statement has to match the port name defined for the ATM OSA-2 adapter. You can configure an ATM OSA-2 adapter using either OSA/SF configuration panels or the IOACMD TSO command. See 2.3.2.4, “Configure ATM OSA-2 adapter using the OSA/SF panels” on page 44 for detailed information on the ATM OSA-2 adapter configuration.

d No IP address associated with the ATM LINK is required, when you configure this statement for SNMP network management only.

Two pairs of the DEVICE and LINK LCS statements have been configured for LEC ports. One port is attached to an ATM token-ring emulated LAN **b**, and the other to an Ethernet ELAN **c**.

2.3.2.4 Configure ATM OSA-2 adapter using the OSA/SF panels

We used the OSA/SF panel to configure the adapter. As the client program, we installed IBM OSA Support Facility program on Windows NT. The following procedure is one way to check the port name configured for an ATM OSA-2 adapter:

1. Customize and start the OSA/SF and IOASRV address space.

   The sample started procedure for IOASRV can be found in IOA.SIOASAMP(IOASRV). The started procedure we are using at ITSO Raleigh is shown below:

```
//*
//*   Sample TCP/IP Server Proc
//*
//SERVER   PROC
//SERVER   EXEC PGM=IOAXTSRV,PARM='8062',REGION=7M,TIME=1440
//STEPLIB  DD  DISP=SHR,DSN=CEE.SCEERUN
//         DD  DISP=SHR,DSN=SYS1.SIOALMOD
//IOALIB   DD  DISP=SHR,DSN=SYS1.SIOALMOD
//SYSPRINT DD  SYSOUT=*,DCB=(RECFM=FBA,LRECL=121,BLKSIZE=121)
//*SYSUDUMP DD  SYSOUT=*
//SYSTCPD  DD  DISP=SHR,DSN=TCPIP.TCPPARMS(TCPD&SYSCLONE.A)   e
```

   e The SYSTCPD DD statement can be used to allocate the TCPIP.DATA data set.

2. Install and start the OSA/SF GUI program on the workstation.

   First you have to establish IP connectivity between the workstation and the host systems that you will use. Then download the program code IOA.AIOAWIN(IOAWINST) from your OS/390 system and save it as the self-extracting file, for example IOAWINST.EXE, on the workstation.

   By starting IOAWINST.EXE, either by double-clicking the IOAWINST object in Windows Explorer, or by entering IOAWINST at a command prompt, the Setup program will install the IBM OSA Support Facility program on your workstation.

   After the installation procedure is completed, you can start it from IBM OSA Support Facility in the Programs view.

3. You can create a connection object with an IP address of the OS/390, and then log into OSA/SF running on OS/390.

4. Change the ATM OSA-2 adapter configuration.

   After selecting an ATM OSA-2 adapter listed in the OSA CHPID's tree view panel, choose **Selected** -> **Configurations...** -> **Configuration lists...**(see Figure 13).

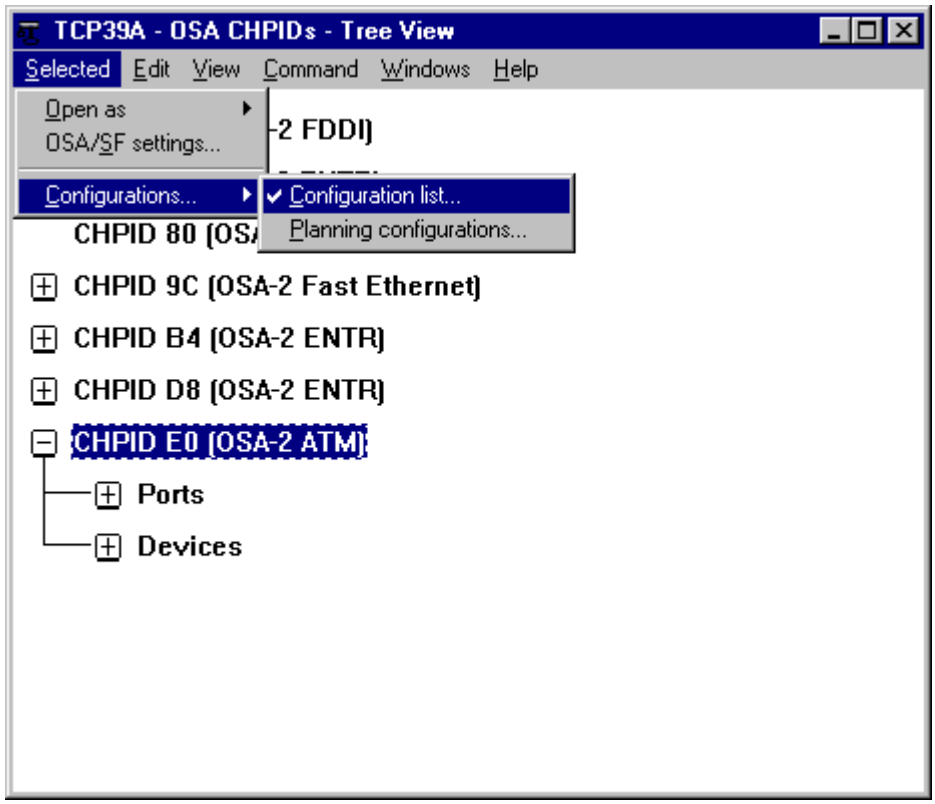


Figure 13. OSA/SF panel: OSA CHPIDs - Tree View

You will see a list of configurations for the selected OSA-2 adapter. Then, select the configuration you are using and press **Change...** (see Figure 14).

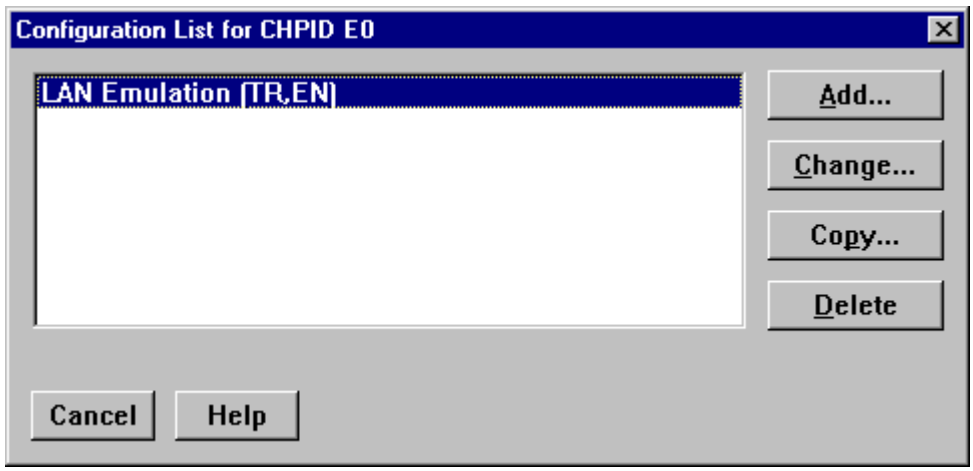


Figure 14. OSA/SF panel: Configuration List

The configuration panels for the selected CHPID will show up. Select ATM LEC Platform (V1.10) in the configured mode list, then press **Change...** (see Figure 15 on page 46).

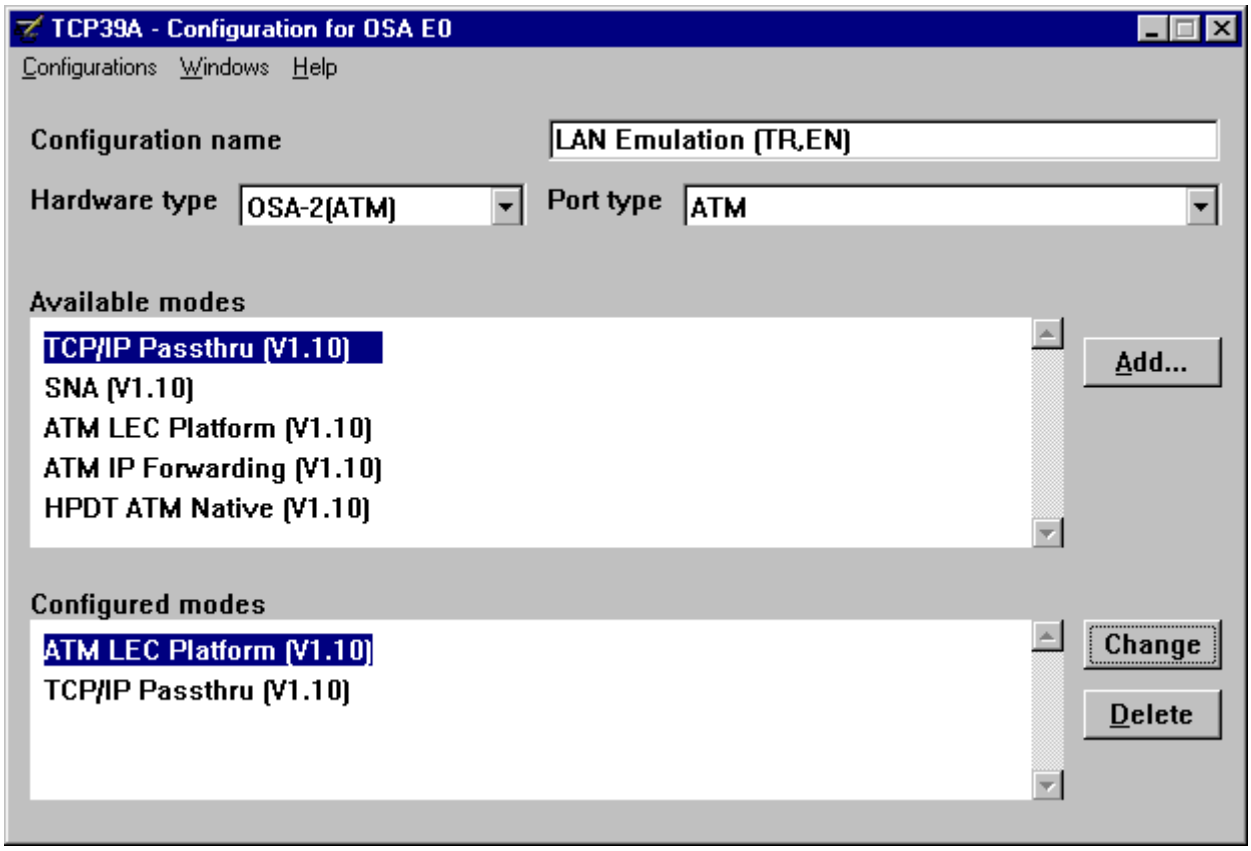


Figure 15. OSA/SF panel: Configuration for the CHPID selected

Finally you will see the ATM LEC Platform Settings panel as shown in Figure 16 on page 47. The port name defined in this panel has to match the port name specified in the DEVICE ATM statement in PROFILE.TCPIP.

For the port name in this panel, the following rules have to be met:

- Specify 1 to 8 of any of the following characters: A through Z in uppercase, @, #, or $ and starting with the second character, 0 through 9.

  Note: Do not use lowercase characters for the port name.

- Do not duplicate a port name across ATM OSA-2s that are to be used in the same mode for the same logical partition.
- Do not specify identical names for the port name and the OSA name of the same ATM OSA-2.
- You must also specify the port name to VTAM in the XCA Port statement and the TRL TRLE statement, if you are using ATM native mode, or SNA.

  Note: If you are using ATM LAN emulation mode only, no VTAM TRLE definition is required.

- If you want SNMP management support, you must also specify the port name in the TCP/IP Device and Link statements.

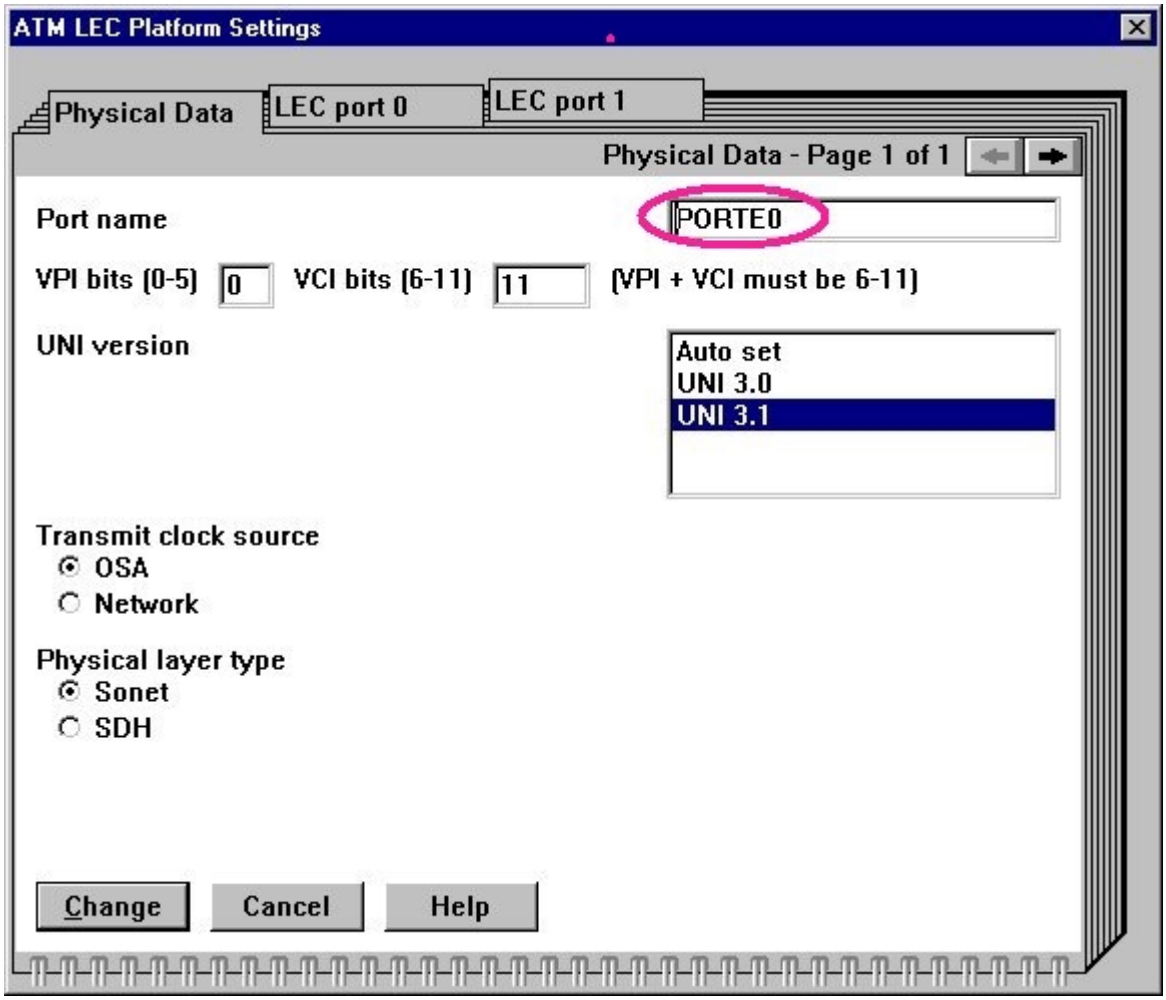


Figure 16. OSA/SF panel: ATM LEC Platform Settings

The port name configured for our ATM OSA-2 adapter is PORTE0, which has also been specified the ATM Device statement of TCPIP.PROFILE. See Figure 12 on page 43 for our TCPIP.PROFILE configuration.

By pressing the **Change** button, the ATM LEC Platform Settings panel (Figure 16) will be closed. If you change the configuration, save and activate it by selecting the corresponding entries in the pull-down menu **Configurations** of the panel shown in Figure 15 on page 46, you must then:

1. Vary on the OSA devices for the affected CHPID offline from all the logical partitions to which the CHPID is defined.
2. Configure the CHPID off, using the `CF CHP(xx),OFFLINE` MVS command.
3. Configure the CHPID on, using the `CF CHP(xx),ONLINE` MVS command.
4. Vary the OSA devices back online to all the logical partitions to which the OSA is defined.

   `xx` is the CHPID for the OSA-2 adapter.

If you use the IOACMD TSO command, the port name has to be configured in an MVS data set. You will see our ATM OSA-2 adapter configuration data set in Appendix E.6, “ATM OSA-2 adapter configuration (CHPID=E0)” on page 284.

For more information on the OSA-2 adapter configuration, refer to *Planning for the S/390 Open Systems Adapter (OSA-1, OSA-2) Feature*, GC23-3870.

2.3.2.5 Starting TCP/IP subagent for ATM OSA-2 adapter management

As we mentioned earlier, the OSA/SF address space must be started before IOASNMP is initialized. In our implementation we used the following startup order:

1. IOAOSASF, which is the OSA/SF address space
2. TCP/IP
3. IOASNMP
4. OSNMPD, which is the SNMP agent

After the TCP/IP subagent has established communication with both IOAOSASF and the SNMP agent, the following message shows up on the MVS console:

```
EZZ3218I SNMP SUBAGENT: CONNECTED TO OSA/SF
```

Then you will be able to retrieve MIB variables that are used to maintain various ATM-related variables. Figure 17 shows an example to retrieve all the entries in ibmMvsAtmLecStatusTable, which represents data available from an OSA port that is running in LAN Emulation mode. These MIB variables are enterprise specific MIB objects defined by IBM. The detailed information on the MIB definitions can be found in /usr/lpp/tcpip/samples/mvstcpip.mib.

```
KAKKY @ RA03:/u/kakky>snmp -t 10 -v -h v3m39a1 walk ibmMvsAtmLecStatusTable
ibmMvsAtmLecPrimaryAtmAddress.16 = '39999999999999000011110221000 4ac20e695f2'h
ibmMvsAtmLecPrimaryAtmAddress.17 = '39999999999999000011110221000 4ac20e695f1'h
ibmMvsAtmLecID.16 = 4
ibmMvsAtmLecID.17 = 3
ibmMvsAtmLecInterfaceState.16 = 7
ibmMvsAtmLecInterfaceState.17 = 7
ibmMvsAtmLecLastFailureRespCode.16 = 1
ibmMvsAtmLecLastFailureRespCode.17 = 1
ibmMvsAtmLecLastFailureState.16 = 1
ibmMvsAtmLecLastFailureState.17 = 1
ibmMvsAtmLecProtocol.16 = 2
ibmMvsAtmLecProtocol.17 = 1
ibmMvsAtmLecVersion.16 = 1
ibmMvsAtmLecVersion.17 = 1
ibmMvsAtmLecTopologyChange.16 = 2
ibmMvsAtmLecTopologyChange.17 = 2
ibmMvsAtmLecConfigServerAtmAddress.16 = '47007900000000000000000000a03e00000100'h
ibmMvsAtmLecConfigServerAtmAddress.17 = '47007900000000000000000000a03e00000100'h
ibmMvsAtmLecConfigSource.16 = 2
ibmMvsAtmLecConfigSource.17 = 2
ibmMvsAtmLecActualLanType.16 = 3
ibmMvsAtmLecActualLanType.17 = 2
ibmMvsAtmLecActualMaxDataFrameSize.16 = 3
ibmMvsAtmLecActualMaxDataFrameSize.17 = 2
ibmMvsAtmLecActualLanName.16 = 'e3d2d96df10000000000000000000000000000000000000000000000000000000000000'h
ibmMvsAtmLecActualLanName.17 = 'c5e3c86df10000000000000000000000000000000000000000000000000000000000000'h
ibmMvsAtmLecAtmAddress.16 = '399999999999990000111101114000 8210000107'h
ibmMvsAtmLecAtmAddress.17 = '399999999999990000111101114000 8210000109'h
ibmMvsAtmLecProxyClient.16 = 2
ibmMvsAtmLecProxyClient.17 = 2
```

Figure 17. Retrieving the entries in LECStatusTable

2.3.3 OMPROUTE subagent configuration

The OMPROUTE subagent extends the set of MIB variables supported by the SNMP agent. The subagent runs as part of OMPROUTE (not as a separate application) and communicates with the SNMP agent using the SNMP Agent Distributed Protocol Interface (DPI) V2.0. The OMPROUTE subagent supports selected MIB objects defined in Request for Comment (RFC) 1850. For a complete list of MIB objects, refer to Appendix C, “Management Information Base (MIB) objects” on page 213.

To configure the OMPROUTE SNMP subagent we have the ROUTESA_CONFIG statement configured in the OPMROUTE configuration data set:

```
ROUTESA_CONFIG ENABLED=YES
               COMMUNITY="MVSsubagent"; 1
```

1 The COMMUNITY name MVSsubagent is coded on the TCPIP.PROFILE SACONFIG statement and defined in the SNMP agent configuration file.

On RA39, we code the following procedure and configuration data sets to initialize OMPROUTE and start the OMPROUTE subagent. The procedure is named OMPROUTA.

```
//OMPROUTA EXEC PGM=BPXBATCH,REGION=4096K,TIME=NOLIMIT,
//         PARM='PGM /usr/lpp/tcpip/sbin/omproute '
//STDOUT   DD PATH='/tmp/omproute.stdout',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//STDERR   DD PATH='/tmp/omproute.stderr',
//         PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//         PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//STDENV   DD DSN=TCPIP.TCPPARMS.R2607(OM39AENV),DISP=SHR  2
//CEEDUMP  DD DUMMY
```

2 The STDENV DD data set contains:

```
RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD39A)'      3
OMPROUTE_FILE=//'TCPIP.TCPPARMS.R2607(OM39ACFG)' 4
```

3 The `RESOLVER_CONFIG` environment variable is configured for the TCPIPA stack. OMPROUTE needs to know the TCPIPJOBNAME coded in TCPIP.DATA.

4 In the OMPROUTE configuration data set we enter the subagent initialization parameters:

```
Area    Area_Number=0.0.0.0
              Stub_Area=NO
              Authentication_type=None;
RouterID=172.16.232.39;
ROUTESA_CONFIG ENABLED=YES
             COMMUNITY="MVSsubagent";    5
OSPF_Interface IP_Address=172.16.102.39
              Name=m392216b
              Cost0=10
              Subnet_mask=255.255.255.0
              MTU=32768;
```

5 This statement activates the OMPROUTE SNMP subagent. The `MODIFY` MVS console command may be used to enable or disable the OMPROUTE subagent

after the subagent is started. To modify the subagent, use the `MODIFY procname,ROUTESA=ENABLE` or the `MODIFY procname, ROUTESA=DISABLE` command.

2.3.4 SLA subagent

The Service Level Agreement (SLA) subagent shipped in CS for OS/390 V2R8 IP provides the following services:

- Retrieves performance monitoring data from the stack and allows monitoring of various QoS characteristics.
- Provides traps for monitored entities (policies/applications) that violate boundaries established using the PolicyMonitorTable MIB.

We ran the SLA subagent on RA28 and RA39 using the started procedure shown below. The sample start procedure is shipped as TCPIP.SEZAINST(PAGTSNMP).

```
//PAGTSNMP PROC
//*
//* SecureWay Communications Server IP
//* SMP/E distribution name: EZAPAGSN
//*
//* 5647-A01 (C) Copyright IBM Corp. 1999.
//* Licensed Materials - Property of IBM
//*
//PAGTSNMP EXEC PGM=PAGTSNMP,REGION=0K,TIME=NOLIMIT,                    1
//       PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c M
//             VSsubagent -t 30'
//*            VSsubagent -d 2 -t 30'
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//*
//* Example of passing parameters to the program (parameters must
//* extend to column 71 and be continued in column 16):
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c p
//*            ublic -P 1234'
//*
//STDENV   DD DSN=TCPIP.TCPPARMS.R2612(PAGT&SYSCLONE.A),DISP=SHR   3
//*
//SYSPRINT DD SYSOUT=*   2
//SYSOUT   DD SYSOUT=*   2
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

1 SLA subagent uses the SNMPv1 protocol to establish communication with the SNMP agent. You must specify the community name for SNMP subagents, which is defined in the SNMP agent configuration file.

2 Output written to stdout and stderr goes to the data set or file specified with SYSPRINT or SYSOUT, respectively. But normally, PAGTSNMP does not write output to stdout or stderr. Instead, output is written to syslogd. When the -o parameter is specified, however, output is written to stdout instead of syslogd.

3 In the MVS data set specified by the STDENV DD, we configured the environment variables as shown below:

```
BROWSE    TCPIP.TCPPARMS.R2612(PAGT39A) - 01.02      Line 000
 Command ===>
******************************** Top of Data ****************
RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD39A)'
TZ=EST5EDT
******************************** Bottom of Data **************
```

2.4 SNMP agent and OS/390 subagents up and running

In this section, we will discuss how SNMP components can be started and show their messages to verify if they have started successfully.

2.4.1 SNMP agent initialization

The SNMP agent may be started from an MVS environment, from the OS/390 UNIX shell, or from the AUTOLOG statement in PROFILE.TCPIP. If started from the OS/390 UNIX shell be sure the name for the UDP port that SNMP uses to receive requests is changed to OMVS.

We started the SNMP agent using the procedure below:

```
//OSNMPD    PROC
//*
//OSNMPD EXEC PGM=EZASNMPD,REGION=4096K,TIME=NOLIMIT,
//       PARM=('POSIX(ON) ALL31(ON)',
//       '/-d 0')
//*
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN    DD DUMMY
//SYSERR   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//CEEDUMP  DD SYSOUT=*              1
//SYSTCPD  DD DSN=TCPIP.TCPPARMS(TCPD39A),DISP=SHR
```

1 OSNMPD must find the name (TCPIPJOBNAME in TCPIP.DATA) with which it should be associated. You can set the environment variable RESOLVER_CONFIG to point to the correct resolver file when multiple INET Physical File Systems (PFS) are started. The SYSTCPD DD card may be used.

The following screen shows the MVS START command and the system responses:

```
S OSNMPD
$HASP100 OSNMPD   ON STCINRDR
IEF695I START OSNMPD   WITH JOBNAME OSNMPD   IS ASSIGNED TO USER TCPIP3
 , GROUP OMVSGRP
$HASP373 OSNMPD   STARTED
IEF403I OSNMPD - STARTED - TIME=11.40.58
EZZ6225I SNMP AGENT: INITIALIZATION COMPLETE
EZZ3218I SNMP SUBAGENT: CONNECTED TO OSA/SF              1
EZZ3202I SNMP SUBAGENT: INITIALIZATION COMPLETE          2
EZZ3221I SNMP SUBAGENT: SET REQUESTS ENABLED
EZZ8101I OMPROUTE SUBAGENT INITIALIZATION COMPLETE       3
```

Figure 18. SNMP agent start messages

1 The TCP/IP subagent has connected to OSA/SF, and the communication to the SNMP agent has been established 2. The OMPROUTE subagent has also established a communication path to the SNMP agent 3 .

When OSNMPD is started with debug level 15, the following log messages will tell you the MVS subagent has connected to the SNMP agent successfully. Those messages will be sent to the syslog daemon and written in its log file.

```
SNMP logging data follows ============================
 Log_type:        snmpLOGrequest_in
    origin:       UDP 127.0.0.1 port 1049
 version:         SNMPv1      1
 community:       MVSsubagent 2
  ..in hex:       4d 56 53 73 75 62 61 67 65 6e 74
 addressInfo:     UDP 127.0.0.1 port 1049   3
 PDUtype:         GetRequest ('a0'h)
 request:         1
 error-status:    noError (0)
 error-index:     0
 varBind oid:
 OBJECT_IDENTIFIER
 1.3.6.1.4.1.2.2.1.1.3.0
 1.3.6.1.4.1.2.2.1.1.3.0
         name:    dpiPathNameForUnixStream.0
         value:
 NULL

 End of SNMP logging data: ============================
 SNMP logging data follows ============================
 Log_type:        snmpLOGresponse_out
    send rc:      0
    destination:  UDP 127.0.0.1 port 1049
 version:         SNMPv1
 community:       MVSsubagent
  ..in hex:       4d 56 53 73 75 62 61 67 65 6e 74
 addressInfo:     UDP 127.0.0.1 port 1049
 PDUtype:         GetResponse ('a2'h)
 request:         1
 error-status:    noError (0)
 error-index:     0
 error-index:     0
 varBind oid:
 OBJECT_IDENTIFIER
 1.3.6.1.4.1.2.2.1.1.3.0
         name:    dpiPathNameForUnixStream.0
         value:
 OCTET_STRING
 (DisplayString)
 /tmp/dpi_socket             4
 End of SNMP logging data: ============================
 EZZ6246I Accepted new DPI UNIX socket connection on fd=7
 pDPIpacket: Major=2, Version=2, Release=0, Id=0, Type=SNMP_DPI_OPEN
 pDPIopen: subagent Identification=1.3.6.1.4.1.2.11.7.2    5
           Description=OS/390 TCP/IP SNMP Subagent         5
           Selected Character Set is 0 (Native)
           timeout=120, max_varBinds=0, password=** NONE **
```

Figure 19. SNMP agent log message (OS/390 TCP/IP SNMP subagent)

1 By sending the SNMPv1 GET request PDU, the SNMP subagent obtains the UNIX stream socket name (4) to communicate with the SNMP agent using AF_UNIX connections. For this request, the community name of MVSsubagent

(2), which has been configured for the SACONFIG statement in TCPIP.PROFILE is used. The IP address that the MVS subagent is using is also shown in this log message (3). This IP address has to match the COMMUNITY definition in the SNMPD.CONF file for the SNMP subagent to connect to the agent. When the subagent opens an AF_UNIX connection with the SNMP agent, the subagent's identification is written (5). Once the stream socket name is received back from the agent as a SNMPv1 response, the community name will not be used for anything else.

In the case of the OMPROUTE subagent and SLA subagent, the identification will be different as shown below:

```
EZZ6246I Accepted new DPI UNIX socket connection on fd=8
 pDPIpacket: Major=2, Version=2, Release=0, Id=0, Type=SNMP_DPI_OPEN
 pDPIopen: subagent Identification=1.3.6.1.4.1.2.11.7.3
           Description=OS/390 TCP/IP OSPF SNMP Subagent
           Selected Character Set is 0 (Native)
           timeout=120, max_varBinds=0, password=** NONE **
```

Figure 20. SNMP agent log message (OS/390 OSPF SNMP subagent)

```
DPIpacket: Major=2, Version=2, Release=0, Id=0, Type=SNMP_DPI_OPEN
DPIopen: subagent Identification=1.3.6.1.4.1.2.11.7.4
         Description=Pagtsnmp SLAPM-MIB DPI Subagent
         Selected Character Set is 0 (Native)
         timeout=120, max_varBinds=0, password=** NONE **
```

Figure 21. SNMP agent log message (OS/390 SLA subagent)

2.4.2 Starting the OMPROUTE subagent

When you start OMPROUTE you should receive the following messages to indicate the successful establishment of the connection between the SNMP agent and the OMPROUTE subagent:

```
S OMPROUTA

EZZ7800I OMPROUTE STARTING
IEE252I MEMBER CTIORA00 FOUND IN SYS1.PARMLIB
EZZ8100I OMPROUTE SUBAGENT STARTING
EZZ7898I OMPROUTE INITIALIZATION COMPLETE
EZZ8101I OMPROUTE SUBAGENT INITIALIZATION COMPLETE
```

Figure 22. OMPROUTE subagent start messages

In the syslogd log, the following messages indicate OMPROUTE successful initialization:

```
EZZ8100I OMPROUTE subagent Starting
EZZ7898I OMPROUTE Initialization Complete
EZZ6246I Accepted new DPI UNIX socket connection on fd=8
EZZ8101I OMPROUTE subagent Initialization Complete
```

Figure 23. The syslogd messages from the OMPROUTE subagent

2.4.3 Starting the service level agreement SNMP subagent

When starting PAGTSNMP, you will see the following message in the MVS console:

```
S PAGTSNMP
$HASP100 PAGTSNMP ON STCINRDR
IEF695I START PAGTSNMP WITH JOBNAME PAGTSNMP IS ASSIGNED TO USER TCPIP3
 , GROUP OMVSGRP
$HASP373 PAGTSNMP STARTED
IEF403I PAGTSNMP - STARTED - TIME=10.34.09
+EZZ8203I PAGTSNMP SUBAGENT: INITIALIZATION COMPLETE
```

Figure 24. SLA subagent start messages

The syslog daemon also writes down some messages from the SLA subagent. If you set the debug flag, you will see detailed information in the syslogd log file.

```
EZZ8222I Pagtsnmp subagent: running as jobname PAGTSNMP
EZZ8217I Pagtsnmp subagent: Using loopback to connect to agent
EZZ6246I Accepted new DPI UNIX socket connection on fd=8
EZZ8203I PAGTSNMP SUBAGENT: INITIALIZATION COMPLETE
```

Figure 25. The syslogd messages from the SLA subagent

When you look at the SNMP agent's log message, you will also be able to make sure the SLA agent has connected to the SNMP agent successfully.

```
F PAGTSNMP,QUERY                                          2
+EZZ8224I PAGTSNMP SUBAGENT: TRACING IS SET TO 2
+EZZ8219I PAGTSNMP SUBAGENT: MODIFY REQUEST COMPLETED

F PAGTSNMP,TRACE,LEVEL=1                                  1
+EZZ8219I PAGTSNMP SUBAGENT: MODIFY REQUEST COMPLETED

F PAGTSNMP,QUERY                                          2
+EZZ8224I PAGTSNMP SUBAGENT: TRACING IS SET TO 1
+EZZ8219I PAGTSNMP SUBAGENT: MODIFY REQUEST COMPLETED
```

Figure 26. Modify command for the SLA subagent

You can change the debug level of the subagent with the `MODIFY` MVS console command (1). To query the current debug level in effect for the SLA subagent, you can also use the `MODIFY` command(2).

2.5 SNMP manager configuration in OS/390 UNIX

The following section describes how to configure the SNMP manager in OS/390 UNIX System Services.

2.5.1 Setting up the OS/390 UNIX SNMP manager

CS for OS/390 V2R8 IP provides two different SNMP manager functions, the `osnmp` and the NetView `snmp` command. At ITSO we have chosen the `osnmp` command for support of the SNMPv3 functions.

Note: Starting with CS for OS/390 V2R8 IP, the `snmp` command is supported in the UNIX shell environment. The command syntax is identical to that of the `osnmp` command.

2.5.1.1 SNMP manager configuration

To issue SNMPv3 requests, the osnmp command needs the OSNMP.CONF file which is used to define target agents and, for SNMPv3, the security parameters to be used in sending requests to them.

The following search order for OSNMP.CONF enables different copies of the file to be used by different users:

1. An HFS file or MVS data set pointed to by the OSNMP_CONF environment variable
2. /etc/osnmp.conf
3. /etc/snmpv2.conf

In our installation, we used the /etc/osnmp.conf HFS file:

```
#--------------------------------------------------------------------------
# Format of entries:
#  winSnmp target   sec   admin password context sec   auth  auth priv  priv
#  Name    Agent    Name                        Level Proto Key  Proto Key
#    1       2       3     4     5         6     7     8     9     10    11
#--------------------------------------------------------------------------
# Community-based security  (SNMPv1 and SNMPv2c)
#-------------------------------------------------
v1       127.0.0.1    snmpv1
v2c      127.0.0.1    snmpv2c
mvs28a   9.24.104.42  snmpv2c
mvs39a   9.24.104.149 snmpv2c
#-------------------------------------------------------------
# User-based Security Model (USM with SNMPV3)
#-------------------------------------------------
v3loc  127.0.0.1    snmpv3 u39  -   -  AuthNoPriv  HMAC-MD5   eb81360f73adf8d6967eaadec03dff45
-   -

v3m39a 9.24.104.149  snmpv3 u39  -   -  AuthNoPriv HMAC-MD5   eb81360f73adf8d6967eaadec03dff45
-   -                                     12
v3m28a 9.24.104.42   snmpv3 u39  -   -  AuthPriv  HMAC-MD5   eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3
v3map   9.24.104.42 snmpv3 u5 password - AuthNoPriv  HMAC-MD5   -              -   -
```

Figure 27. Example /etc/osnmp.conf file

Note: In order to show the complete statements, the lines are wrapped to the next line. Be sure to code your statements on only one line.

The definitions are:

1 The label by which osnmp identifies an entry in this table. This name is specified in the -h command option.

2 The target agent IP address or name.

3 Specifies the administrative model adopted by the agent. Valid values are snmpv1, snmpv2c, and snmpv3.

4 The user name. This name must be known by the agent.

5 The password used to generate the key (see 2.2.4, “Key generation for authentication and privacy” on page 31). If a password is specified, it is used to automatically generate any needed keys and the authKey and privKey fields below are ignored.

Note: The use of passwords instead of keys in this configuration file is *not* recommended, as storing passwords in this file is less secure than using keys.

6 The SNMP contextName to be used at the target agent. Sicne the CS for OS/390 SNMP agent does not support multiple contexts, set this field to a single dash (-).

7 The security level requested.

8 The SNMP authentication protocol used (see 2.2.4, “Key generation for authentication and privacy” on page 31).

9 The authentication key generated for this user.

10 Specifies that encryption (privacy) will be used to communicate with this agent.

11 The privacy key generated for this user.

12 The manager will communicate with this server using authentication and privacy checking.

In *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513, you will find a detailed description of all parameters available for the osnmp configuration.

If you want to use textual names for MIB objects which are not defined in any compiled MIB, then you can define them to the `osnmp` command in an /etc/mibs.data HFS file. All of the objects in the list in Appendix C, “Management Information Base (MIB) objects” on page 213, are in the compiled MIB. A sample of the /etc/mibs.data file is installed as HFS file /usr/lpp/tcpip/samples/mibs.data.

2.5.1.2 Working with the OS/390 UNIX SNMP manager

The following sample screens should give you an idea of what you can do with the `osnmp` command.

You can find the `osnmp` command syntax and parameter description in the *OS/390 SecureWay Communications Server IP User’s Guide*, GC31-8514.

The Management Information Base (MIB) objects are listed in Appendix C., “Management Information Base (MIB) objects” on page 213. The table shows by object the object descriptor, object identifier, if the object is supported by the agent or subagent, where the object is defined (RFC number, subagent MIB, etc.), and the access allowed (read-only or read/write).

You can find the MIB definitions in the HFS directory /usr/lpp/tcpip/samples.

You might need to debug your SNMP `osnmp` command environment. To do this, you must activate debug levels 1 to 4. The log information is normally displayed on your screen. To capture the log information in a file, you can redirect output into an HFS file as shown in the following screen:

```
MARTIN @ RA39:/u/martin>snmp -d 1 -h v3m39a walk system > /tmp/osnmp.log 2>&1
```

All user screen data is recorded in the HFS file specified in the redirection command (>). If the file you specified does not exist, the system creates this file automatically.

Throughout our test we used SNMPv3 authentication, so the -h parameter identified the label name defined in /etc/osnmp.conf.

The following screen shows the system display for our SNMP agent on system MVS39. We provided some of this information in the agent configuration HFS file /etc/osnmpd.data:

```
MARTIN @ RA39:/u/martin>osnmp -h v3m39a walk system
1.3.6.1.2.1.1.1.0 = SNMPv3 agent version 1.0 with DPI version 2.0
1.3.6.1.2.1.1.2.0 = 1.3.6.1.4.1.2.3.13
1.3.6.1.2.1.1.3.0 = 848700
1.3.6.1.2.1.1.4.0 = Tatsuhiko Kakimoto
1.3.6.1.2.1.1.5.0 = CS for OS/390 V2R8 on SYSTEM39
1.3.6.1.2.1.1.6.0 = ITSO Raleigh, IBM
1.3.6.1.2.1.1.7.0 = 76
1.3.6.1.2.1.1.8.0 = 3100
1.3.6.1.2.1.1.9.1.2.1 = 1.3.6.1.4.1.2.11.7.1
1.3.6.1.2.1.1.9.1.2.2 = 1.3.6.1.4.1.2.11.7.2
1.3.6.1.2.1.1.9.1.2.3 = 1.3.6.1.4.1.2.11.7.3
1.3.6.1.2.1.1.9.1.3.1 = OS/390 SNMP Agent
1.3.6.1.2.1.1.9.1.3.2 = OS/390 TCP/IP SNMP Subagent
1.3.6.1.2.1.1.9.1.3.3 = OS/390 TCP/IP OSPF SNMP Subagent
1.3.6.1.2.1.1.9.1.4.1 = 100
1.3.6.1.2.1.1.9.1.4.2 = 1700
1.3.6.1.2.1.1.9.1.4.3 = 3100
```

Figure 28. Display of the system MIB

By using the -v option for the osnmp command, you can display the output from a request as a textual name instead of the MIB object identifier. This option might help you to understand the MIB information more easily.

2.5.1.3 Using SNMPv1/SNMPv2 GET command

The SNMP agent shipped with CS for OS/390 V2R8 IP supports the use of SNMPv1, SNMPv2, and SNMPv3 concurrently. To implement the SNMPv1 and/or SNMPv2 support, as with the user-based security model, both the SNMP manager and agent must be configured for SNMPv1 and SNMPv2. In our installation, OSNMP.CONF, which is the SNMP manager configuration file, has community definitions as follows:

```
#-----------------------------------------------
# Community-based security  (SNMPv1 and SNMPv2c)
#-----------------------------------------------
1           2          3
v1        127.0.0.1    snmpv1
v2c       127.0.0.1    snmpv2c
mvs03av1  9.24.104.113 snmpv1
mvs39av1  9.24.104.149 snmpv1
mvs39a    9.24.104.149 snmpv2c
```

1 winSNMPname: An administrative name by which the winSNMP code used by osnmp can locate an entry in this configuration file. This is the target SNMP agent for the osnmp command. There is no default value, but the name is specified with the -h option.

2 targetAgent: Host name or IP address of the node of the target agent.

3 admin: Specifies the administrative model supported by the targetAgent. Valid values are:

- snmpv1 - community-based SNMPv1 security
- snmpv2c - community-based SNMPv2 security
- snmpv3 - user-based SNMPv3 security

In the SNMP agent's SNMPD.CONF configuration file for the view-based access control model (VACM) definition, the community name used on an SNMPv1 and SNMPv2 request must be defined to the SNMP agent with a community entry, as well as the corresponding VACM_GROUP, VACM_ACCESS, and VACM_VIEW definitions. We have the following groups/views defined:

```
#-----------------------------------------
VACM_GROUP group3 SNMPv1  publicv1  -
VACM_GROUP group3 SNMPv2c publicv2c -

VACM_GROUP group4 SNMPv1  MVSsubagent  -
VACM_GROUP group4 SNMPv2c MVSsubagent  -
            4      5         6
```

4 groupName is the group name for VACM. The groupName must be specified; there is no default value.

5 securityModel indicates the SNMP security model for this entry. When an SNMP message comes in, the securityModel and the securityName are used to determine the group to which the user (or community) represented by the securityName belongs. Valid values are:

- SNMPv1: indicates community-based security using SNMPv1 message processing.
- SNMPv2c: indicates community-based security using SNMPv2c message processing.
- USM: indicates the user-based security model.
- - (dash): indicates the default value of USM.

6 securityName indicates a community name for the community-based security model that is used in either SNMPv1 or SNMPv2c. For the USM security model, specify a USM user name for this group. Valid values are 1-32 characters.

Also define the VACM_ACCESS entries for the groups defined:

```
# VACM_ACCESS entries
# Format is:
#           group context context security security  read     write        notify  storage
#            Name  Prefix  Match   Level    Model    View     View         View     Type
#-------------------------------------------------------------------------------------------------
VACM_ACCESS  group3  -  7   -    noAuthNoPriv SNMPv1  bigView   bigView       bigView -
VACM_ACCESS  group3  -      -    noAuthNoPriv SNMPv2c bigView   bigView       bigView -

VACM_ACCESS  group4  -      -    noAuthNoPriv SNMPv1  subagentView   -             -    -
VACM_ACCESS  group4  -      -    noAuthNoPriv SNMPv2c subagentView   -             -    -
```

7 VACM_ACCESS definitions for group3. See Figure 10 on page 36 for view configurations.

The following demonstrates the use of an SNMPv1 query from OS/390 UNIX on RA28. See Figure 29 on page 59 for the OS/390 UNIX SNMP manager configuration information on RA28.

```
MARTIN @ RA28:/u/martin>snmp -v -h mvs39av1 -c publicv1 walk system
sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0
sysObjectID.0 = 1.3.6.1.4.1.2.3.13
sysUpTime.0 = 585700
sysContact.0 = Tatsuhiko Kakimoto
sysName.0 = CS for OS/390 V2R8 on SYSTEM39
sysLocation.0 = ITSO Raleigh, IBM
sysServices.0 = 76
sysORLastChange.0 = 100
sysORID.1 = 1.3.6.1.4.1.2.11.7.1
sysORID.2 = 1.3.6.1.4.1.2.11.7.2
sysORID.3 = 1.3.6.1.4.1.2.11.7.3
sysORDescr.1 = OS/390 SNMP Agent
sysORDescr.2 = OS/390 TCP/IP SNMP Subagent
sysORDescr.3 = OS/390 TCP/IP OSPF SNMP Subagent
sysORUpTime.1 = 0
sysORUpTime.2 = 100
sysORUpTime.3 = 100
```

2.5.1.4 Using SNMPv3 GET command

As with SNMPv1 and SNMPv2 security, both the manager and agent must be configured to allow for USM-based security for SNMPv3. Figure 27 on page 55 details the configuration for the SNMP manager on RA39. USM security may be configured as AuthNoPriv, AuthPriv or noAuthnoPriv. For privacy, see Table 4 on page 17 for the encryption feature that must be installed before privacy (encryption) can be implemented. The USM userName must be configured on both manager and agent. For details on configuring the SNMP manager (osnmp.conf) and SNMP agent (snmpd.conf) files, refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513.

For our configuration of the SNMP manager on RA28, we used the OSNMP.CONF as follows:

```
#------------------------------------------------
# Community-based security  (SNMPv1 and SNMPv2c)
#------------------------------------------------
v1       127.0.0.1   snmpv1
v2c      127.0.0.1   snmpv2c
mvs03av1  9.24.104.113 snmpv1
mvs39av1  9.24.104.149 snmpv1
mvs39a  9.24.104.149 snmpv2c
#------------------------------------------------------------
# User-based Security Model (USM with SNMPV3)
# MLE - 01/21/2000  Modified v3m39a to use encryption (AuthPriv)
#------------------------------------------------
v3m39a  9.24.104.149 snmpv3 u28 - - AuthPriv   HMAC-MD5 9f1626506dbd7540c8ce526070fb69f9 DES
7865f0c012464308a5a4deead83903c3
v3m39a1 9.24.104.149 snmpv3 u28 - - AuthNoPriv HMAC-MD5 9f1626506dbd7540c8ce526070fb69f9 -   -
v3m28a  9.24.104.42  snmpv3 u28 - - AuthNoPriv HMAC-MD5 9f1626506dbd7540c8ce526070fb69f9 -   -
```

Figure 29. OS/390 UNIX SNMP manager configuration on RA28

Below are two examples of SNMPv3 manager request commands and SNMPv3 agent responses. The first uses SNMPv3 with authentication only and the second is SNMPv3 with authentication and privacy as shown below:

```
                                          1
MARTIN @ RA28:/u/martin>osnmp -v -h v3m39a1 get sysDescr.0
sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0

                                          2
MARTIN @ RA28:/u/martin>osnmp -v -h v3m39a get sysDescr.0
sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0
```

Figure 30. SNMP request using SNMPv3 with different security levels

1 This is the SNMPv3 request with authentication only. Host name v3m39a1 designated by the -h parameter specifies HMAC-MD5 security with a configured non-localized authentication key as shown in our example OSNMP.CONF above.

2 An SNMPv3 request with authentication and privacy. The host name in this example, v3m39a is configured as AuthPriv and includes a non-localized authentication key and the DES parameter followed by a non-localized privacy key. All SNMP data transmitted is encrypted.

For information on creating authentication and privacy keys, see the section on Creating User Keys in *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513.

2.5.2 Receive SNMP traps using the SNMP manager

SNMP agent trap information is coded in the SNMPD.CONF using the NOTIFY entries, TARGET_ADDRESS and TARGET_PARAMETERS parameters. See Figure 10 on page 36 for the installation configuration of the SNMP.CONF on RA39 at ITSO.

To get the trap information, we used the following started task on the system RA03:

```
//OSNMPTRP PROC
//OSNMPTRP EXEC PGM=BPXBATCH,REGION=30M,TIME=NOLIMIT,
//            PARM='PGM /bin/osnmp -d 0 -p 16200 trap'
//*
//STDERR DD PATH='/tmp/osnmptrp.err',PATHOPTS=(OWRONLY,OCREAT,OAPPEND),
//       PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//STDOUT DD PATH='/tmp/osnmptrp.out',PATHOPTS=(OWRONLY,OCREAT,OAPPEND),
//       PATHMODE=(SIRUSR,SIWUSR,SIRGRP,SIWGRP)
//STDENV DD DSN=TCPIP.TCPPARMS.R2612(SNM03TRP),DISP=SHR
//*
//CEEDUMP  DD DUMMY
```

In the MVS data set specified by the STDENV DD, we configured the environment variables as shown below:

```
RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD03A)'
TZ=EST5EDT
```

To receive only trap packets sent from SNMP agents in our OS/390 systems, we decided to run the SNMP manager on UDP port number 16200 rather than the well-known UDP port number 162.

The trap information received by the osnmp command is written in an HFS file, /tmp/osnmptrp.out. An example of trap information is shown below:

```
Display of SNMPv1 trap:          1
community:       publicv1        4
enterprise oid:  1.3.6.1.4.1.2.3.13
        .....    ibmTcpIpMvs
agentAddress:    172.16.232.39 ('ac10e827'h)   5
generic-trap:    coldStart ('00000000'h)       6
specific-trap:   0 ('00000000'h)
time-stamp:      0 - 0.00 seconds

Display of SNMPv2 trap with SNMPV2C security    2
community:       publicv2c                       4
agent-address:   9.24.104.149 ('09186895'h)      5
generic-trap:    coldStart ('00000000'h)         6
specific-trap    0 ('00000000'h)
time-stamp:      0 - 0.00 seconds
Trap varBinds:
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.6.3.1.1.4.3.0
        name:    snmpTrapEnterprise.0
        value:   OBJECT_IDENTIFIER 1.3.6.1.4.1.2.3.13
        .....    ibmTcpIpMvs

Display of SNMPv2 trap with USM security         3
security-name:   u03
security-level:  AuthNoPriv
security-model:  USM
contextEngineID: '000000020000000009186895'h
context-name:    NULL
agent-address:   9.24.104.149 ('09186895'h)      5
generic-trap:    coldStart ('00000000'h)         6
specific-trap    0 ('00000000'h)
time-stamp:      0 - 0.00 seconds
Trap varBinds:
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.6.3.1.1.4.3.0
        name:    snmpTrapEnterprise.0
        value:   OBJECT_IDENTIFIER 1.3.6.1.4.1.2.3.13
        .....    ibmTcpIpMvs
```

Figure 31. Traps received by osnmp

Since we configured the SNMP agent to send three versions of traps (SNMPv1 1, SNMPv2c 2 and SNMPv3 3), the osnmp command receives the coldStart (6) traps in each version. You will see the agent's IP address (5) and the community name (4) contained in the traps, and other information shown above.

Note: Traps can also be received and displayed using the NetView snmp command and the NetView NCCF panel. See Appendix A, "Working with the NetView SNMP command" on page 165 for a description of the NetView snmp command.

Chapter 3. Managing OS/390 TCP/IP using SNMP

The OS/390 UNIX `osnmp` command provides the SNMP manager function from the OS/390 UNIX shell and can be used to create or alter many MIB objects. For example, The `osnmp set` command may be used to change the configurations of the SNMP agent or the TCP/IP stack. This chapter describes how to monitor and/or update the MIB values using the `osnmp` command.

3.1 MIB objects supported by CS for OS/390 IP

The Management Information Base (MIB) defines the objects that may be managed for each layer in the TCP/IP protocol. The MIB is a hierarchical structure. It is extendable, from a required minimum base. Some *branches* in the MIB *tree* are required to be implemented on all platforms. Some branches of the tree are optional, but broadly used. Other branches of the tree contain data that is specific to a platform; these are called enterprise-specific.

There are two versions: MIB-I and MIB-II. MIB-I was defined in RFC 1156, and is now classified as a historic protocol with a status of not recommended.

Currently there are about 10,000 standards-based MIB objects defined and many more enterprise-specific objects in the MIB tree. The SNMP agent and subagents shipped as part of OS/390 support about 850 MIB objects.

Table 10. SNMP MIB support RFC summary

MIB Support	RFC Number
Interfaces	RFC 2233
IP	RFC 2011
IP Forwarding Table	RFC 1354 (obsoleted by RFC 2096)
ICMP	RFC 2011
TCP	RFC 2012
UDP	RFC 2013
SNMP	RFC 1907, RFC 2271-RFC 2275
OSPF	RFC 1850
IP over ATM	RFC 2320
DPI	RFC 1592
ATM Interface	RFC 1695 (obsoleted by RFC 2515)
Service Level Agreement Management	RFC 2758
IBM 3172	IBM 3172 Enterprise Specific MIB
IBM Remote Ping	IBM TCP/IP MVS Enterprise Specific MIB
IBM TCP/IP MVS	IBM TCP/IP MVS Enterprise Specific MIB

Table 10 summarizes the RFCs that define the MIB objects supported by CS for OS/390 V2R8 IP. Please refer to the RFCs listed for the complete definition. The Service Level Agreement Performance Monitor (SLAPM) MIB is defined in RFC

2758, which was published in February 2000, however its implementation on OS/390 V2R8 IP is based on draft-white-slapm-mib-03.txt, January 1999. The draft RFC document is also provided with the file name slapm.txt in the /usr/lpp/tcpip/samples directory.

The /usr/lpp/tcpip/samples directory also contains the agent capabilities statement, which indicates which MIBs are supported by the SNMP agent and subagents shipped with CS for OS/390 V2R8 IP, including any variations on our support, such as read/write objects supported as read-only. The /usr/lpp/tcpip/samples directory also contains the enterprise-specific MIB definitions. See Appendix C, “Management Information Base (MIB) objects” on page 213 for a list of all supported MIB objects and the subagents shipped with CS for OS/390 V2R8.

Some MIB objects may change their values dynamically in response to the SNMP SET request sent from an SNMP manager that may be running either on the local or a remote system. Some MIB objects may not only have their value updated, but the objects themselves may be created by the SNMP requests received.

Table 11 describes some of the MIB objects that may be updated, created, or deleted by the SNMP agents through the SNMP operations.

Table 11. MIB object that may be updated dynamically

MIB object name (MIB object identifier)	**Descriptions**
SNMP agent	
sysContact (1.3.6.1.2.1.1.4)	The textual identification of the contact person for the system being managed. The value may be configured in OSNMPD.DATA.
sysName (1.3.6.1.2.1.1.5)	The textual name for this managed system. The value may be configured in OSNMPD.DATA.
sysLocation (1.3.6.1.2.1.1.6)	The physical location of this managed system. The value may be configured in OSNMPD.DATA.
snmpEnableAuthenTraps (1.3.6.1.2.1.11.30)	Indicates whether the SNMP entity is permitted to generate authenticationFailure traps. The value may be configured in OSNMPD.DATA.
snmpTargetAddrTable (1.3.6.1.6.3.12.1.2) 1	This table contains several objects that are equivalent to the parameters defined for the TARGET_ADDRESS statements in SNMPD.CONF. By creating and/or deleting the entries for the SnmpTargetAddrEntry object, you can add and/or delete the trap destination dynamically.
snmpTargetParamsTable (1.3.6.1.6.3.12.1.3) 1	This table contains several objects that are equivalent to the parameters defined for the TARGET_PARAMETERS statements in SNMPD.CONF. You can create/delete the entries in this table, SnmpTargetParamsEntry, through the SNMP operations.

MIB object name (MIB object identifier)	Descriptions
usmUserTable (1.3.6.1.6.3.15.1.2.2) 1	This table maintains authentication and privacy information for each user. The entry consists of several objects that are equivalent to the parameters defined for the USM_USER statements in SNMPD.CONF. By creating/deleting/updating the entries in this table, a user may add/remove/change his or her security level.
vacmAccessTable (1.3.6.1.6.3.16.1.4) 1	This table maps a group name, security information, a context, and a message type, such as read, write, or notification, into a MIB view. An entry has the corresponding information to that defined in the VACM_ACCESS statement in SNMPD.CONF. By creating a new object, a new MIB view can be defined.
TCP/IP subagent	
ifAdminStatus (1.3.6.1.2.1.2.2.1.7)	The desired state of the interface. The possible state is either up(1), down(2), or testing(3). By changing these values, the interfaces may be started and/or stopped via the SNMP operation.
ipForwarding (1.3.6.1.2.1.4.1)	Indicates whether this managed node is acting as an IP router in respect to the forwarding of datagrams received by, but not addressed to, this node.
tcpConnState (1.3.6.1.2.1.6.13.1.1)	The state of this TCP connection. By setting this object to the value deleteTCB(12), which is the only value you may set, the corresponding connection on the managed node will be terminated immediately.
ibmMvsTcpKeepAliveTimer (1.3.6.1.4.1.2.6.19.2.2.2.4)	TCP Keepalive timer, expressed in minutes. A value of 0 deactivates the timer. The default value is 120 (minutes).
ibmMvsTcpReceiveBufferSize (1.3.6.1.4.1.2.6.19.2.2.2.5)	The default TCP Receive buffer size, expressed in bytes. The default value is 16384 bytes.
ibmMvsTcpSendBufferSize (1.3.6.1.4.1.2.6.19.2.2.2.6)	The default TCP Send buffer size, expressed in bytes. The default value is 16384 bytes.
ibmMvsSoMaxConn (1.3.6.1.4.1.2.6.19.2.2.2.10)	Maximum number of socket connections.
ibmMvsUdpSendBufferSize (1.3.6.1.4.1.2.6.19.2.2.2.21)	The default UDP send buffer size, expressed in bytes. The default value is 65535 bytes.
ibmMvsUdpRecvBufferSize (1.3.6.1.4.1.2.6.19.2.2.2.22)	The default UDP send buffer size, expressed in bytes. The default value is 65535 bytes.
ibmMvsPathMtuDscEnabled (1.3.6.1.4.1.2.6.19.2.2.2.22)	The indication of whether the PATHMTUDISCOVERY IP option is enabled. The possible values are true(1) or false(2).
ibmMvsMultipathType (1.3.6.1.4.1.2.6.19.2.2.2.34)	The type of multipath processing currently active. The types are defined under the MULTIPATH parameter on the IPCONFIG Profile statement. The possible values are either nomultipath(0), perconnection(2), or perpacket(3).
SLA subagent	

MIB object name (MIB object identifier)	Descriptions
slapmPolicyTrapEnable (1.3.6.1.3.88.1.1.7)	Indicates whether slapmPolicyProfileDeleted and slapmPolicyMonitorDeleted notifications should be generated by this system.
slapmPolicyMonitorTable (1.3.6.1.3.88.1.2.2) [1]	This table provides a method of monitoring the effect of SLA policy being used at a system. The entries in this table have to be created using the SNMP SET requests.
[1] These objects are not accessible objects; that is they are not leaves but intermediate nodes.	

For the complete list of MIB objects that are supported by CS for OS/390 V2R8 IP, refer to Appendix C, “Management Information Base (MIB) objects” on page 213.

3.1.1 What you can do by monitoring the MIB values

The SNMP framework provides very powerful methods to manage IP networks. It allows network administrators to address various networking management related issues. Since these operations are done using the SNMP protocol, a network administrator can reside anywhere in the IP network. In other words, he or she no longer has to log into the target systems to maintain network nodes.

3.1.1.1 Traffic monitoring

By monitoring the amount of data transmitted over a router’s interfaces, an administrator can monitor how many bytes of data have been transmitted between IP subnetworks in a particular period of time. You can also monitor the traffic size over an interface of an IP host that is running as an application server. The following are examples of MIB objects that can be used for this monitoring:

- ifInOctets: gives the total number of octets received on the interface, including framing characters.
- ifOutOctets: gives the total number of octets transmitted out of the interface, including framing characters.
- sysUpTime: provides a count in one hundredths of a second of how long the SNMP agent has been running, that is how long the device has been up and running in most cases. This value is useful to determine the time differences from the previous collection and whether previous counter information is valid. Therefore this value should be retrieved with each collection.

The receive utilization of an interface can be calculated from the formula below using these MIB values:

```
utilization(%) = (delta(ifInOctets) x 8 / (ifSpeed x delta(sysUpTime))) x 100
```

where `delta` means the difference between two MIB values retrieved at the different points in time. The transmit utilization also can be calculated using the ifOutOctets values.

Note: the ifSpeed MIB value gives the transmission rate in bits/second.

3.1.1.2 Problem determination

Some MIB values are very useful for problem determination in IP networks, such as performance problems or lack of connectivity.

Examples of MIB objects that can be used to determine problems in the data-link layer are:

- ifInErrors, ifOutErrors: give the number of inbound/outbound frames that were discarded due to errors.
- ifInDiscards, ifOutDiscards: give the number of inbound/outbound frames that were discarded due to resource limitations.

Examples of MIB objects that can be used to determine problems in the IP layer are:

- ipInHdrErrors: gives the number of IP packets that were discarded because of errors in their IP headers.
- ipInUnknownProtos: indicates the number of IP packets that were received successfully but discarded because of either unknown or unsupported protocol.
- ipInDiscards, ipOutDiscards: give the number of inbound/outbound IP packets that were discarded due to resource limitations.
- ipReasmFails: provides the number of failures detected by the IP re-assembly algorithm (for whatever reason: timed out, errors, etc).

The following MIB objects are useful to determine problems related to the TCP or UDP protocol:

- tcpRetransSegs: gives the number of TCP segments retransmitted.
- udpInErrors: gives the number of UDP datagrams discarded for reasons other than the lack of an application at the destination port.

3.1.1.3 TCP/IP stack management

By changing particular MIB values, a TCP/IP stack configuration can be altered without recycling. Some of the operations supported are:

- Change the IP forwarding attributes.
- Change the logic of the equal-cost multipath activated.
- Drop an existing TCP connection established to a remote IP host.
- Alter the buffer size allocated for each TCP connection and/or UDP association.

See 3.2, “Manage the TCP/IP stack using SNMP operations” on page 68, for our sample implementations.

3.1.1.4 SNMP agent configuration

The SNMPv3 framework allows you to change the SNMP agent’s configuration dynamically. Some example operations supported are:

- Change the security keys for SNMP managers.
- Add new users, or SNMP managers.
- Change the group definition.

See 3.3, “SNMP agent configuration with SNMP” on page 75, for our sample implementations.

3.1.1.5 Performance monitoring

The SLA subagent maintains the SLA Performance Monitor MIB (SLAP-MIB) that gives various performance information for each policy installed in a TCP/IP stack. Some possible monitorings are:

- Monitor the throughput.
- Monitor the transmission delay.
- Monitor the amount of data that meet a policy.

Refer to 3.4, “Monitoring performance using MIB information and traps” on page 86, for more information and the sample implementation.

3.2 Manage the TCP/IP stack using SNMP operations

The TCP/IP stack, OMPROUTE and the policy agent configuration can be updated by changing the MIB values managed by them. Table 11 on page 64 describes some MIB objects that can be dynamically configured. For a complete list of MIB objects that have an allowed access of read-write (R/W), read-create (R/C) or write-only (W/O) see Appendix C, “Management Information Base (MIB) objects” on page 213.

3.2.1 TCP/IP stack configuration using SNMP

Several MIB objects managed by the TCP/IP SNMP subagent may be changed to alter the TCP/IP stack configuration. In our example we will demonstrate how the SNMP manager can change several settings configured in the IPCONFIG section of TCPIP.PROFILE without using the `OBEYFILE` command.

3.2.1.1 Change the OSPF multipath configuration

In /usr/lpp/tcpip/samples/mvstcpip.mi2, the ibmMvsMultipathType object is defined as follows:

```
ibmMvsMultipathType    OBJECT-TYPE
SYNTAX  INTEGER {
                   nomultipath(0),
                   perconnection(2),
                   perpacket(3)
                 }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
      "The type of Multipath processing currently
       active.  The types are defined under the MULTIPATH
       parameter on the IPCONFIG Profile statement."
DEFVAL { nomultipath }
::= { ibmTcpipMvsSystem 34 }
```

The value of the ibmMvsMultipathType MIB object is initially set by the NOMULTIPATH or MULTIPATH statement in TCPIP.PROFILE. For details on configuring the IPCONFIG statements in TCPIP.PROFILE, refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513. In our

IPCONFIG statement, we coded MULTIPATH. If MULTIPATH is specified without any keywords, the default is PERCONNECTION.

```
D TCPIP,TCPIPA,N,CONFIG
EZZ2500I NETSTAT CS V2R8 TCPIPA 062
           :
           :
IP Configuration Table:
Forwarding: Yes    TimeToLive: 00060  RsmTimeOut:  00015
FireWall:   00000  ArpTimeout: 01200  MaxRsmSize:  65535
IgRedirect: 00001  SysplxRout: 00001  DoubleNop:   00000
StopClawEr: 00001  SourceVipa: 00001  VarSubnet:   00001
MultiPath:  Conn 1 PathMtuDsc: 00000
           :
```

Figure 32. IPCONFIG portion of NETSTAT CONFIG

1 A portion of the NETSTAT CONFIG display in Figure 32 shows multipath perconnection is configured on RA39.

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a get ibmMvsMultipathType.0
ibmMvsMultipathType.0 = 2 2
```

Figure 33. SNMP display of the ibmMvsMultipathType MIB object

2 In Figure 33 an SNMP manager display of the MIB object shows a value of 2 or perconnection status.

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a set ibmMvsMultipathType.0 3
ibmMvsMultipathType.0 = 3  3
```

Figure 34. Using SNMP to change the value of the bmMvsMultipathType MIB object

3 Changing the ibmMvsMultipathType MIB object value from 2 to 3, will change the multipath from perconnection to perpacket.

```
D TCPIP,TCPIPA,N,CONFIG
EZZ2500I NETSTAT CS V2R8 TCPIPA 062
           :
           :
IP Configuration Table:
Forwarding: Yes    TimeToLive: 00060  RsmTimeOut:  00015
FireWall:   00000  ArpTimeout: 01200  MaxRsmSize:  65535
IgRedirect: 00001  SysplxRout: 00001  DoubleNop:   00000
StopClawEr: 00001  SourceVipa: 00001  VarSubnet:   00001
MultiPath:  Pkt 4  PathMtuDsc: 00000
```

Figure 35. Portion of NETSTAT CONFIG showing OMPROUTE multipath perpacket

4 Multipath perpacket is now enabled. In the MVS system log, the following message is displayed:

```
EZZ3229I SNMP SUBAGENT: MULTIPATH PERPACKET SUPPORT IS ENABLED
```

3.2.1.2 Alter the IP forwarding definition

In another example, we may change the behavior of IP forwarding through SNMP by setting the ibmMvsIpForwarding MIB object. The status of this MIB object

indicates whether this managed node is acting as an IP router in respect to the forwarding of datagrams received by, but not addressed to, this node.

The following description of the ibmMvsIpForwarding can be found in /usr/lpp/tcpip/samples/mvstcpip.mi2:

```
ibmMvsIpForwarding     OBJECT-TYPE
SYNTAX  INTEGER {
                   notForwarding(1),
                   forwardingNoMultipath(2),
                   forwardingMultipathPkt(3)
                 }
MAX-ACCESS  read-write
STATUS current
DESCRIPTION
      "The indication of whether this entity is acting as an IP
       router in respect to the forwarding of datagrams received
       by, but not addressed to, this entity.  If this entity is
       acting as a router, this object also indicates what type
       of forwarding is in effect.  The value of this object
       is originally determined by the TCP/IP Profile
       ASSORTEDPARMS or IPCONFIG statements.  A value of
       notForwarding(1) indicates that this entity is not
       forwarding datagrams.  A value of forwardingNoMultipath(2)
       indicates that this entity is forwarding datagrams using
       the first available route.  A value of
       forwardingMultipathPkt(3) indicates that this entity,
       when forwarding datagrams, will select a route on a
       round-robin basis from a multipath routing list.
          This object is related to the ipForwarding object
       from RFC2011.  If ipForwarding is set to notForwarding,
       this object will also be set to notForwarding.  If
       ipForwarding is set to forwarding, this object will be
       set to forwardingNoMultipath.  If the value of this MIB
       object is changed, the value of ipForwarding will be
       changed accordingly."
DEFVAL { forwardingNoMultipath }
::= { ibmTcpipMvsSystem 35 }
```

The value for IP forwarding is initially set by the NODATAGRAMFWD or DATAGRAMFWD statement in the IPCONFIG statement in TCPIP.PROFILE. If DATAGRAMFWD is specified without any subparameters, the default is NOFWDMULTIPATH.

```
IP Configuration Table:
Forwarding: Yes  5 TimeToLive: 00060  RsmTimeOut:  00015
FireWall:   00000  ArpTimeout: 01200  MaxRsmSize:  65535
IgRedirect: 00001  SysplxRout: 00001  DoubleNop:   00000
StopClawEr: 00001  SourceVipa: 00001  VarSubnet:   00001
MultiPath:  Pkt    PathMtuDsc: 00000
```

Figure 36. Portion of NETSTAT CONFIG showing OMPROUTE IP forwarding with no multipath

5 At ITSO we coded DATAGRAMFWD in the IPCONFIG statement.

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a get ibmMvsIpForwarding.0
ibmMvsIpForwarding.0 = 2  6
```

Figure 37. SNMP display of ibmMvsIpForwarding

6 The value of 2 for the MIB object ibmMvsIpForwarding indicates forwardingNoMultipath.

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a set ibmMvsIpForwarding.0 3
ibmMvsIpForwarding.0 = 3   7
```

Figure 38. Using SNMP to change the value of the ibmMvsIpForwarding MIB object

7 Changing the ibmMvsIpForwarding MIB object value from 2 to 3, will change the IP forwarding from forwardingNoMultipath to forwardingMultipathPkt.

```
IP Configuration Table:
Forwarding: Pkt  8 TimeToLive: 00060  RsmTimeOut:  00015
FireWall:   00000  ArpTimeout: 01200  MaxRsmSize:  65535
IgRedirect: 00001  SysplxRout: 00001  DoubleNop:   00000
StopClawEr: 00001  SourceVipa: 00001  VarSubnet:   00001
MultiPath:  Pkt    PathMtuDsc: 00000
```

Figure 39. Portion of NETSTAT CONFIG showing OMPROUTE IP forwarding perpacket

8 Datagram forwarding has been changed from forwarding datagrams using the first available route (forwardingNoMultipath) to forwarding datagrams using a route on a round-robin basis from a multipath routing list (forwardingMultipathPkt). In the MVS system log, the following message is displayed:

```
EZZ3230I SNMP SUBAGENT: IP FORWARDING FWDMULTIPATH PERPACKET SUPPORT IS ENABLED
```

3.2.2 Manage TCP connections using SNMP

You can maintain the status of a TCP connection using the SNMP operations. The TCP connection table (tcpConnTable) defined as an object in the TCP group maintains information about the existing TCP connection on the IP host being managed. Each entry in the table is indexed by local and remote IP addresses and TCP port numbers (see Table 12).

Table 12. The object in tcpConnTable

Object	Description	Type	Access
tcpConnState	TCP connection state	Integer	read-write
tcpConnLocalAddress	Local IP address	IpAddress	read-only
tcpConnLocalPort	Local TCP port number	Integer	read-only
tcpConnRemAddress	Remote IP address	IpAddress	read-only
tcpConnRemPort	Remote TCP port number	Integer	read-only

The tcpConnState MIB object is defined in RFC 2012 as follows:

```
tcpConnState OBJECT-TYPE
    SYNTAX      INTEGER {
                    closed(1),
                    listen(2),
                    synSent(3),
                    synReceived(4),
                    established(5),
                    finWait1(6),
                    finWait2(7),
                    closeWait(8),
                    lastAck(9),
                    closing(10),
                    timeWait(11),
                    deleteTCB(12)
                }

    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
            "The state of this TCP connection.

            The only value which may be set by a management station is
            deleteTCB(12). Accordingly, it is appropriate for an agent
            to return a `badValue' response if a management station
            attempts to set this object to any other value.
            If a management station sets this object to the value
            deleteTCB(12), then this has the effect of deleting the TCB
            (as defined in RFC 793 ) of the corresponding connection on
            the managed node, resulting in immediate termination of the
           connection.
```

Transmission Control Block (TCB) is a data structure to maintain the TCP connection information, such as the local and remote socket numbers. A TCB is created for each TCP connection when it is established, and deleted when the TCP connection is terminated, or the TCP connection state changes to CLOSED in the TCP state transition diagram. In contrast, by deleting a TCB, you can close the associated TCP connection. For more information on the TCB and TCP state transition diagram, refer to *RFC 793 - Transmission Control Protocol*.

By using the `osnmp` command, we can delete the TCB of an existing TCP connection by setting the tcbConnState MIB object value to 12. Therefore, you can maintain the TCP connection information on remote systems, and close a TCP connection established between two remote systems. Note that, the `NETSTAT DROP` command can close a TCP connection addressed to a local TCP/IP host only.

For our example, an FTP connection is established from a Windows NT workstation to RA39, and we issue the `osnmp` command on a different OS/390 image, which is RA03 in our configuration.

```
C:\>ftp mvs39a
Connected to mvs39a.itso.ral.ibm.com.
220-FTPDA1 IBM FTP CS V2R8 at MVS39A.itso.ral.ibm.com, 14:59:48 on 2000-03-02.
220 Connection will close if idle for more than 60 minutes.
User (mvs39a.itso.ral.ibm.com:(none)): martin
331 Send password please.
Password:
230 MARTIN is logged on.  Working directory is "MARTIN.".
ftp>
```

Figure 40. Establish FTP connection to RA39

We can now display the tcpConnState MIB object:

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a walk tcpConnState
tcpConnState.0.0.0.0.21.0.0.0.0.0 = 2   1
tcpConnState.0.0.0.0.23.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.760.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.1025.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.1031.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.10007.0.0.0.0.0 = 2
tcpConnState.9.24.104.149.21.9.24.106.40.1459 = 11  2
tcpConnState.9.24.104.149.21.9.24.106.40.1462 = 5   3
tcpConnState.127.0.0.1.760.127.0.0.1.1030 = 5
tcpConnState.127.0.0.1.1025.127.0.0.1.1028 = 5
tcpConnState.127.0.0.1.1026.127.0.0.1.1027 = 5
tcpConnState.127.0.0.1.1027.127.0.0.1.1026 = 5
tcpConnState.127.0.0.1.1028.127.0.0.1.1025 = 5
tcpConnState.127.0.0.1.1030.127.0.0.1.760 = 5
```

Figure 41. Displaying the tcpConnState MIB object

Figure 41 shows the tcpConnState MIB objects and their values. The different values represent the state of this TCP connection.

1 tcpConnState value of 2 states the connection is in the listen status such as FTPD listening on port 21.

2 This is an example of the tcpConnState MIB object. As you can see, the object is indexed by local IP address, local TCP port number, remote IP address, and remote TCP port number. Using the index information, a TCP connection can be identified easily.

In our example, we opened an FTP connection to RA39 and immediately closed it. The value of 11 represents timeWait indicating the connection was already terminated however the TCB connection information was still temporarily maintained.

3 This is our active FTP connection to RA39. tcpConnState value of 5 indicates the connection is established.

These two connections have been established by FTP clients running on the same workstation, whose IP address is 9.24.106.40. Each FTP client is assigned a unique TCP port number, which is 1459 for the first TCP connection and 1462 for the second one, so that the TCP connection information can be maintained in the table.

By setting the TCP connection state to 12, a TCP connection can be terminated. In our example, to drop the FTP connection, we will set the value of tcpConnState.9.24.104.149.21.9.24.106.40.1462 to 12.

The state value 12 is the only value you can set using the SNMP operations. Setting the value of this MIB object to anything else will result in an error. Figure 42 shows both setting tcpConnState to an invalid value and setting it to deleteTCB (12).

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a set tcpConnState.9.24.104.149.21.9.24.106.40.1462 5
tcpConnState.9.24.104.149.21.9.24.106.40.1462 = wrongValue                    4
EZZ3330I SET request failed.  No values changed.

MARTIN @ RA28:/u/martin>snmp -v -h v3m39a  set tcpConnState.9.24.104.149.21.9.24.106.40.1462 12
tcpConnState.9.24.104.149.21.9.24.106.40.1462 = 12     5
```

Figure 42. Dropping TCP/IP connection by changing tcpConnState

In Figure 42, an attempt to change the tcpConnState value to anything other than 12 results in an error 4. The SNMP manager successfully sets the tcpConnState MIB object for the FTP connection to 12 which deletes the connection 5.

```
C:\>ftp mvs39a
Connected to mvs39a.itso.ral.ibm.com.
220-FTPDA1 IBM FTP CS V2R8 at MVS39A.itso.ral.ibm.com, 14:59:48 on 2000-03-02.
220 Connection will close if idle for more than 60 minutes.
User (mvs39a.itso.ral.ibm.com:(none)): martin
331 Send password please.
Password:
230 MARTIN is logged on.  Working directory is "MARTIN.".
ftp> dir
Connection closed by remote host. 6
ftp>
```

Figure 43. FTP connection after changing the tcpConnState value

In the FTP connection, we receive notification 6 that TCP/IP on the OS/390 system has closed the connection as shown in Figure 43. A display of the tcpConnState MIB object shows the established FTP connection has been closed as we see in Figure 44.

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a walk tcpConnState
tcpConnState.0.0.0.0.21.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.23.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.760.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.1025.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.1031.0.0.0.0.0 = 2
tcpConnState.0.0.0.0.10007.0.0.0.0.0 = 2
tcpConnState.9.24.104.149.21.9.24.106.40.1462 = 11   7
tcpConnState.127.0.0.1.760.127.0.0.1.1030 = 5
tcpConnState.127.0.0.1.1025.127.0.0.1.1028 = 5
tcpConnState.127.0.0.1.1026.127.0.0.1.1027 = 5
tcpConnState.127.0.0.1.1027.127.0.0.1.1026 = 5
tcpConnState.127.0.0.1.1028.127.0.0.1.1025 = 5
tcpConnState.127.0.0.1.1030.127.0.0.1.760 = 5
```

Figure 44. Dropping TCP/IP connection by changing tcpConnState

7 The FTP connection is no longer active.

3.3 SNMP agent configuration with SNMP

The `osnmp` command can be used to change the user-based security model or view-based access model configurations of the SNMPv3 agent.

3.3.1 Update the secret keys for authentication and/or privacy

SNMPv3 user-based security model (USM) enhances security between an SNMPv3 manager and agent through the use of authentication and privacy (encryption) keys. These security keys can be dynamically changed on the SNMPv3 agent using the `osnmp` command. The key updates are also written to the SNMPD.CONF file of the SNMPv3 agent. By updating the SNMPv3 agent authentication and privacy keys dynamically, security updates can be implemented based on the requirements of the enterprise.

In order to accomplish this, we perform the following steps:

1. Issue the `pwchange` command.
2. Update the usmUserAuthKeyChange or usmUserPrivKeyChange MIB object on the SNMP agent using keyChange data generated from the `pwchange` command.
3. Issue the `pwtokey` command to create the new security key for the SNMP manager.
4. Edit the SNMP manager's OSNMP.CONF file and update security key.
5. Verify successful request/response from manager to agent.

The `pwchange` command is an OS/390 UNIX shell command that is provided to facilitate dynamic key changes for USM users. Authentication and privacy keys can be changed while the SNMP agent is running by setting the usmUserAuthKeyChange or usmUserPrivKeyChange values.

All objects in the MIB tree have an instance (or index) associated with them. For objects for which there can be only one copy, the index is simply 0, as in sysDescr.0. However many objects can have multiple copies.

Authentication and privacy information for each user is maintained in the usmUserTable MIB table. All rows in this table are indexed by an SNMP engine identifier and a user name. Table 13 lists all the MIB objects defined in usmUserTable. RFC 2574 provides more information on the user-based security model (USM) for SNMPv3.

Table 13. The usmUserTable definition

Object	Description	Type	Access
usmUserEngineID	An SNMP engine's administratively-unique identifier. This value will simply be the snmpEngineID value of the local agent. However, if a local agent can generate SNMP request messages, then this can be a remote snmpEngineID that this entity may send SNMP requests to.	SnmpEngine ID	N/A

Object	Description	Type	Access
usmUserName	The name of a user in textual string. The usmUserEngineID along with the usmUserName is used to index into this table.	SnmpAdmin String	N/A
usmUserSecurityName	The name of a user in security model independent format. On CS for OS/390 V2R8 IP, the value must be the same as the usmUserName object.	SnmpAdmin String	R/O
usmUserCloneFrom	A pointer to an existing row in this usmUserTable. When a new user is created, this object must be set to point to an existing row, and the security objects, usmUserAuthProtocol and usmUserPrivProtocol, are copied from the cloned row.	RowPointer	R/C
usmUserAuthProtocol	An indication of the authentication protocol to use. None, HMAC-MD5-96, and HMAC-SHA-96 are supported.	Autonomous Type	R/C
usmUserAuthKeyChange	This object is used to change a user's authentication key.	KeyChange	R/C
usmUserOwnAuthKeyChange	Behaves exactly as usmUserAuthKeyChange, except that only the user who owns this row can set the value.	KeyChange	R/C
usmUserPrivProtocol	An indication of the privacy protocol to use. None and CBC-DES are supported.	Autonomous Type	R/C
usmUserPrivKeyChange	This object is used to change a user's privacy key.	KeyChange	R/C
usmUserOwnPrivKeyChange	Behaves exactly as usmUserPrivKeyChange, except that only the user who owns this row can set the value.	KeyChange	R/C
usmUserPublic	This object can be used to validate that a set operation to change a user's secret authentication and/or privacy key succeeded.	OCTET STRING	R/C
usmUserStorageType	The storage type for this row.	StorageType	R/C
usmUserStatus	The status of this row. A row can not be made active until usmUserCloneFrom is set. Further, a newly created row for a user who employs authentication and/or privacy, cannot be made active until the corresponding usmUserAuthKeyChange and/or usmUserPrivKeyChange has been set.	RowStatus	R/C
N/A: not-accessible R/O: read-only R/C: read-create			

Table 14 shows the structure of the index for the usmUserPrivKeyChange MIB object, which is indexed by the engine ID and user name.

See Figure 45 for the display of usmUserPrivKeyChange at ITSO.

Table 14. Index structure for MIB usmUserPrivKeyChange

Object name	Engine ID index		user Name index	
usmUserPrivKeyChange	length (1byte)	Engine ID	length(1 byte)	User name in ASCII

In our example, we will change the privacy key for user u28 configured on the RA39 SNMP agent. Refer to Figure 10 on page 36 to view the original RA39

agent configuration file. For more information on the pwchange command, refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513.

In Figure 45, we display the usmUserPrivKeyChange and usmUserSecurityName MIB objects. By displaying the usmUserSecurityName, the USM user name will be displayed in the SNMP response. This may help to identify which MIB object to change without converting the ASCII representation.

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a walk usmUserPrivKeyChange
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = NULL
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = NULL
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = NULL
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = NULL
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = NULL
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.10
8 = NULL              1          2                3         4

MARTIN @ RA28:/u/martin>snmp -v -h v3m39a walk usmUserSecurityName
usmUserSecurityName.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = u03
usmUserSecurityName.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = u28
usmUserSecurityName.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = u39  5
usmUserSecurityName.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = u03n
usmUserSecurityName.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = u28n
usmUserSecurityName.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108
 = initial
```

Figure 45. Example SNMP display of usmUserPrivKeyChange and usmUserSecurityName

1 One-byte field containing the number of bytes in the engine ID.

2 The engine ID of the SNMP agent (see the section “SNMPD.BOOTS” on page 29 for more information on the engine ID).

3 One-byte field containing the number of bytes in the USM user name.

4 The USM user name in ASCII representation.

5 It may be helpful to display the usmUserSecurityName MIB for the textual representation of the USM user name.

The pwchange command creates the value (a keyChange value) to be used on an SNMP SET command; it does not change the key directly itself. A keyChange value is a hexadecimal string twice as long as the key to be changed. Dynamic key changes are done by setting keyChange values rather than storing the keys directly in MIB objects because sending the new keys themselves "over the wire" would not be secure. The keyChange value hexadecimal number is derived by knowing the old password from which the original key was derived and providing a new password to be used for the new key. For additional security, a keyChange value incorporates a random component. For that reason, a repeated pwchange command with the same input parameters will generate different keyChange values as output.

```
MARTIN @ RA28:/u/martin>pwchange -p HMAC-MD5 -u priv privpass newprivpass 9.24.104.149
                                 6          7     8        9            a
Dump of 32 byte HMAC-MD5 privKey keyChange value:
  d356ca149e76333fd81be8492f514e41
  f0fa7606cc46677fc62ee573a3b5e159
                                          b
MARTIN @ RA28:/>snmp -v -h v3m39a set usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56
                     \'d356ca149e76333fd81be8492f514e41f0fa7606cc46677fc62ee573a3b5e159\'h

usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 =
'd356ca149e76333fd81be8492f514e41f0fa7606cc46677fc62ee573a3b5e159'h
```

Figure 46. SNMP command to change USM user privacy key

6 You can specify the protocol for which the keychange values should be generated.

7 Key usage. In our example, we are changing the privacy key. Without this parameter, this command creates an authentication keychange value. Note that there is no difference between a keyChange value generated for authentication and a keychange value generated for privacy.

8 , 9 The original and new USM passwords.

a IP address of SNMP agent on RA39.

In our example b , we will change the USM privacy password for user u28 on the SNMP agent on RA39 with the `snmp set` command. Note that the entire command from the UNIX shell will encompass more than the command buffer. You can use a back slash (\) as the last character prior to filling the buffer and continue typing the command on the next line. This will concatenate the segments into one command.

After the command is issued, the following messages from the SNMP agent are written to the syslog on RA39.

```
snmpagent[33554479]: IDSTMVS.S@DW0016.SOURCE.S@DYNBAS(310)
snmpagent[33554479]: Dyn.Conf: remote_snmp_usm_set invoked
snmpagent[33554479]: IDSTMVS.S@DW0016.SOURCE.S@DYNBAS(311)
snmpagent[33554479]: Dyn. Conf: operation : 177 Status : 1
snmpagent[33554479]: IDSTMVS.S@DW0016.SOURCE.S@DYNBAS(382)
snmpagent[33554479]: Dyn.Conf: Modify Received
snmpagent[33554479]: IDSTMVS.S@DW0016.SOURCE.S@DYNBAS(310)
snmpagent[33554479]: Dyn.Conf: remote_snmp_usm_set invoked
snmpagent[33554479]: IDSTMVS.S@DW0016.SOURCE.S@DYNBAS(311)
snmpagent[33554479]: Dyn. Conf: operation : 174 Status : 1
snmpagent[33554479]: IDSTMVS.S@DW0016.SOURCE.S@DYNBAS(382)
snmpagent[33554479]: Dyn.Conf: Modify Received
   :
snmpagent[33554479]: IDSTMVS.S@DW0249.SOURCE.S@DYNFUP(432)
snmpagent[33554479]: Dyn. Conf: Updated Config file          c
snmpagent[33554479]: IDSTMVS.S@DW0249.SOURCE.S@DYNFUP(445)
snmpagent[33554479]: Dyn. Conf: After updating config file
```

c indicates that the TCP/IP SNMP subagent has updated the privacy key definition for user u28 in the SNMPD.CONF configuration file.

The USM_USER portion of the SNMP SNMPD.CONF configuration file has been updated for user u28 as follows:

```
#-------------------------------------------------------------------------------------------
# USM_USER entries
# Format is:
#  userName engineID authProto authKey privProto privKey keyType storageType
#
##-------------------------------------------------------------------------------------------
USM_USER u03  - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  DES 61a5dfe8a7277f1f982edce11e9fc633 L -
USM_USER  u03n - none -   - d -  L -
USM_USER u28 000000020000000009186895 HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97 DES da17ff9cb12620fe00aa5fed515b41da  L nonVolatile
USM_USER  u28n - none -   -   -  L -                                                e                    f
USM_USER  u39 - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  -   -  L -
#USM_USER  u28a - HMAC-MD5 4a93130df02117a66dd80cbf1cf0ee9c  -   -  L -
```

Figure 47. SNMP agent configuration file updated by TCP/IP subagent

In the SNMPD.CONF file the localized privacy key **e** has been updated for USM user u28. Several other values have been updated in the file. The engine ID **d** replaces the '-' (dash) with the default value (the local SNMP agent's engine ID) and the storageType **f** has been updated with the default value of nonVolatile.

Note: The MVS MODIFY command may be used to change the interval at which dynamic configuration changes are written to the SNMPD.CONF configuration file. The interval may be set between zero and ten minutes. The default value is five minutes. See Modifying SNMP Agent Parameters in *OS/390 SecureWay Communications Server IP User's Guide*, GC31-8514

On RA39, the SNMP agent has now been dynamically updated with a new privacy key for USM user u28. The `osnmp` command (SNMP manager) uses the privacy key found on an entry in the OSNMP.CONF configuration file. It needs to correlate with the privacy key specified on a USM_USER entry for that user in the agent SNMPD.CONF configuration file. The SNMP manager communicates with RA39 as USM u28 and the OSNMP.CONF must be updated with the new privacy key. The SNMP manager configuration can only be updated manually. In our example, the SNMP manager is on RA28. Figure 48 shows the `pwtokey` command used to generate the non localized privacy key to update the SNMP manager's OSNMP.CONF file on RA28. For details on the `pwtokey` command see *OS/390 SecureWay Communications Server IP User's Guide*, GC31-8514.

```
MARTIN @ RA28:/u/martin>pwtokey -p HMAC-MD5 -u priv newprivpass 9.24.104.149
Display of 16 byte HMAC-MD5 privKey:
  fa8180fe93bb703fbb9be95f700df071

Display of 16 byte HMAC-MD5 localized privKey:
  da17ff9cb12620fe00aa5fed515b41da
```

Figure 48. UNIX pwtokey command for updated privacy password

On the SNMP manager OSNMP.CONF file on RA28 we update as follows:

```
#-------------------------------------------------
# Community-based security  (SNMPv1 and SNMPv2c)
#-------------------------------------------------
v1       127.0.0.1   snmpv1
v2c      127.0.0.1   snmpv2c
mvs03av1  9.24.104.113 snmpv1
mvs39av1  9.24.104.149 snmpv1
mvs39a  9.24.104.149 snmpv2c
itso2216  9.24.104.1 snmpv1
itso2210  9.24.104.3 snmpv1
```

```
#-----------------------------------------------

# User-based Security Model (USM with SNMPV3)
#-----------------------------------------------
v3m39a  9.24.104.149 snmpv3 u28  - - AuthPriv  HMAC-MD5 eb81360f73adf8d6967eaadec03dff45 DES
fa8180fe93bb703fbb9be95f700df071 g
v3m39a1 9.24.104.149 snmpv3 u28 - - AuthNoPriv HMAC-MD5 eb81360f73adf8d6967eaadec03dff45 - -
v3m39aN 9.24.104.149 snmpv3 u28n - - noAuthNoPriv  - - - -
v3m28a  9.24.104.42  snmpv3 u28  - - AuthNoPriv HMAC-MD5 eb81360f73adf8d6967eaadec03dff45  -   -
```

g The non-localized password generated with `pwtokey`

After the SNMP manager OSNMP.CONF file is updated, you can verify the changes:

```
MARTIN @ RA28:/u/martin>snmp -v -h v3m39a get sysDescr.0
sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0
```

Figure 49. Verify pwchange update

3.3.2 Define a new user for SNMPv3 security

Using the `osnmp` command we can define a new USM user on the SNMP agent. After defining the new user, the SNMP agent's SNMPD.CONF file is dynamically updated with the new information.

To create a new user for the SNMPv3 agent, the following steps were taken:

1. Set the usmUserStatus MIB value to createAndWait (5).
2. Set the usmUserCloneFrom object to another row from which you want to clone.
3. Set the usmUserAuthKeyChange value to a new keyChange value using a new authentication password. If the new user will not use authentication, set the usmUserAuthProtocol MIB value to usmNoAuthProtocol.
4. Set the usmUserPrivKeyChange MIB value to the keyChange value based on the new privacy key. If the new user will not use the privacy protocol, set the usmUserPrivProtocol MIB value to usmNoPrivProtocol.
5. Set the usmUserStatus to active (1).
6. Set the group for the new user, and issue the SNMP SET command to update the vacmGroupName object for the new user.
7. Activate the group status by changing the user's vacmSecurityToGroupStatus value to one (active)

In our example, we defined user u04 for the SNMP agent running on RA39 and set the VACM_GROUP for this user to group1. The new user's security information is copied from the entries of user u03, which has been defined to use authentication protocol only (AuthNoPriv). The authentication key for the user u03 was generated using the password `authpass`, which has to be used to create the keyChange values for the new user.

Figure 50 shows the status of USM users before user 04 has been defined.

```
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk usmUserStatus
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108 = 1
```

Figure 50. usmUserStatus setting before defining a new user

The usmUserStatus is defined with the type RowStatus, which provides a standard way to manage rows in a table. The possible values are:

- active (1)
- notInService (2)
- notReady (3)
- createAndGo (4)
- createAndWait (5)
- destroy (6)

For more information on the RowStatus object, refer to *RFC 2579 - Textual Conventions for SMIv2*.

First we created the new row for the new user by cloning it from an existing row.

By setting the value of the usmUserCloneFrom object indexed with the new user's name to point to the existing row, a row for the new user is created and the security information is also copied from the cloned-from user.

```
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 set usmUserCloneFrom.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 \
> usmUserCloneFrom.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51                    1

usmUserCloneFrom.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 1.3.6.1.6.3.15.1.2.2.1.4.12.0.0.0.2.0.0.0.0.9.24.104.149.

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk usmUserStatus
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 3      2
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108 = 1

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk usmUserPrivProtocol
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = 1.3.6.1.6.3.10.1.2.1
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 1.3.6.1.6.3.10.1.2.1    3
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = 1.3.6.1.6.3.10.1.2.2
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = 1.3.6.1.6.3.10.1.2.1
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = 1.3.6.1.6.3.10.1.2.1
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = 1.3.6.1.6.3.10.1.2.1
usmUserPrivProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108 = 1.3.6.1.6.3.10.1.2.1

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk usmUserAuthProtocol
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = 1.3.6.1.6.3.10.1.1.2
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 1.3.6.1.6.3.10.1.1.2    4
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = 1.3.6.1.6.3.10.1.1.2
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = 1.3.6.1.6.3.10.1.1.2
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = 1.3.6.1.6.3.10.1.1.1
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = 1.3.6.1.6.3.10.1.1.1
usmUserAuthProtocol.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108 = 1.3.6.1.6.3.10.1.1.2
```

Table 15. Create the raw for the user being added

1 We created the row for the new user, u04, by cloning it from user u03.

2 This is the usmUserStatus table setting. The usmUserStatus entry for the new user, u04, has been defined with the status of notReady (3). To accomplish the task of defining a new user, the status must be set to active (1).

3, 4 You will see that the security information, such as usmUserAuthProtocol, has been copied from the clone-from user. These objects are defined in RFC 2574 as shown in Table 16.

Table 16. Security protocol definition in RFC 2274

Security Protocol	Value	Object Identifier
Authentication		
None	usmNoAuthProtocol	1.3.6.1.6.3.10.1.1.1
HMAC-MD5-96	usmHMACMD5AuthProtocol	1.3.6.1.6.3.10.1.1.2
HMAC-SHA-96	usmHMACSHAAuthProtocol	1.3.6.1.6.3.10.1.1.3
Privacy		
None	usmNoPrivProtocol	1.3.6.1.6.3.10.1.2.1
CBC-DES	usmDESPrivProtocol	1.3.6.1.6.3.10.1.2.2

Then we generate the keyChange value based on the authentication keys for the clone-from user 5 and the new user 6 using the pwchange utility.

```
                                        5         6
KAKKY @ RA03:/u/kakky>pwchange -p HMAC-MD5 authpass newauthpass 9.24.104.149
Dump of 32 byte HMAC-MD5 authKey keyChange value:
  003af769f9449d304d1d025f8045f934
  237c4f7edbac777cd805880d7fdcfd60


KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 set usmUserAuthKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 \
 \'003af769f9449d304d1d025f8045f934237c4f7edbac777cd805880d7fdcfd60\'h        7
usmUserAuthKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 =
'003af769f9449d304d1d025f8045f934237c4f7edbac777cd805880d7fdcfd60'h

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 set usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 \'\'h
usmUserPrivKeyChange.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = NULL
```

Figure 51. Set the authentication and privacy keys for a new user

7 The keyChange value has to be set to the usmUserAuthKeyChange object for the new user, so that the user's secret authentication can be defined in the SNMP agent configuration. This object must be set after the new user is cloned from an existing user.

8 Since we do not use the privacy protocol for SNMPv3 communication, the secret key for privacy has been set to NULL.

By setting both the usmUserAuthKeyChange and usmUserPrivKeyChange values to appropriate values, the usmUserStatus changes to notInService (2) 9, which means the row is ready and waiting.

```
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk usmUserStatus
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 2          9
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108 = 1

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 set usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 1                  a

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk usmUserStatus
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.51 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.48.52 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.50.56 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.3.117.51.57 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.48.51.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.4.117.50.56.110 = 1
usmUserStatus.12.0.0.0.2.0.0.0.0.9.24.104.149.7.105.110.105.116.105.97.108 = 1
```

Figure 52. Making the usmUserStatus active

a Then you can issue the SNMP SET command to activate the usmUserStatus.

After adding the new user in USM, you have to define to which group he or she belongs, so that the access control in the MIB tree is implemented through the View-Based Access Control Model (VACM), which associates users to MIB views.

In our sample implementation, the new user u04 will be defined as a member of the existing group group1. The MIB view for members of group1 is defined in the SNMPD.CONF configuration file.

```
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk vacmGroupName          b
vacmGroupName.1.8.112.117.98.108.105.99.118.49 = group3
vacmGroupName.1.8.151.164.130.147.137.131.165.241 = group3
vacmGroupName.1.11.77.86.83.115.117.98.97.103.101.110.116 = group4
vacmGroupName.1.11.212.229.226.162.164.130.129.135.133.149.163 = group4
vacmGroupName.2.9.112.117.98.108.105.99.118.50.99 = group3
vacmGroupName.2.9.151.164.130.147.137.131.165.242.131 = group3
vacmGroupName.2.11.77.86.83.115.117.98.97.103.101.110.116 = group4
vacmGroupName.2.11.212.229.226.162.164.130.129.135.133.149.163 = group4
vacmGroupName.3.3.117.48.51 = group1
vacmGroupName.3.3.117.50.56 = group1           c
vacmGroupName.3.3.117.51.57 = group1
vacmGroupName.3.4.117.48.51.110 = group1
vacmGroupName.3.4.117.50.56.110 = group1
vacmGroupName.3.7.105.110.105.116.105.97.108 = initial
                                                                    d
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 set vacmGroupName.3.3.117.48.52 group1
vacmGroupName.3.3.117.48.52 = group1

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk vacmGroupName
vacmGroupName.1.8.112.117.98.108.105.99.118.49 = group3
                 :
vacmGroupName.2.11.212.229.226.162.164.130.129.135.133.149.163 = group4
vacmGroupName.3.3.117.48.51 = group1
vacmGroupName.3.3.117.48.52 = group1                    e
vacmGroupName.3.3.117.50.56 = group1
vacmGroupName.3.3.117.51.57 = group1
                 :
```

Figure 53. Define the new user as a group member

b Before defining the user's membership there is no vacmGroupName MIB object associated with the user. This object is indexed by the user name (c) and the value is the group name in textual format to which he or she belongs.

d By issuing the SNMP SET command, a user can be defined as a member of a particular group. We did not define a new group for this user, but defined him or her as a member of an existing group. You have to set the value of the vacmGroupName MIB object indexed by the user name being defined (d) to a group name he or she will join.

If the SET request completes successfully, the MIB object for the new user will be created with the group name specified (e).

Then you have to activate the vacmSecurityToGroupStatus MIB object for the user, which is defined with the type RowStatus, by setting its value to one (active).

```
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk vacmSecurityToGroupStatus
vacmSecurityToGroupStatus.1.8.112.117.98.108.105.99.118.49 = 1
vacmSecurityToGroupStatus.1.8.151.164.130.147.137.131.165.241 = 1
vacmSecurityToGroupStatus.1.11.77.86.83.115.117.98.97.103.101.110.116 = 1
vacmSecurityToGroupStatus.1.11.212.229.226.162.164.130.129.135.133.149.163 = 1
vacmSecurityToGroupStatus.2.9.112.117.98.108.105.99.118.50.99 = 1
vacmSecurityToGroupStatus.2.9.151.164.130.147.137.131.165.242.131 = 1
vacmSecurityToGroupStatus.2.11.77.86.83.115.117.98.97.103.101.110.116 = 1
vacmSecurityToGroupStatus.2.11.212.229.226.162.164.130.129.135.133.149.163 = 1
vacmSecurityToGroupStatus.3.3.117.48.51 = 1
vacmSecurityToGroupStatus.3.3.117.48.52 = 2          f
vacmSecurityToGroupStatus.3.3.117.50.56 = 1
vacmSecurityToGroupStatus.3.3.117.51.57 = 1
vacmSecurityToGroupStatus.3.4.117.48.51.110 = 1
vacmSecurityToGroupStatus.3.4.117.50.56.110 = 1
vacmSecurityToGroupStatus.3.7.105.110.105.116.105.97.108 = 1

                                                          g
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 set vacmSecurityToGroupStatus.3.3.117.48.
vacmSecurityToGroupStatus.3.3.117.48.52 = 1

KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a1 walk vacmSecurityToGroupStatus
vacmSecurityToGroupStatus.1.8.112.117.98.108.105.99.118.49 = 1
                        :
vacmSecurityToGroupStatus.2.11.212.229.226.162.164.130.129.135.133.149.163 = 1
vacmSecurityToGroupStatus.3.3.117.48.51 = 1
vacmSecurityToGroupStatus.3.3.117.48.52 = 1                              h
vacmSecurityToGroupStatus.3.3.117.50.56 = 1
vacmSecurityToGroupStatus.3.3.117.51.57 = 1
                        :
```

Figure 54. Activating vacmSecurityToGroupStatus for a user

f When the user's membership is defined, the status is set to notInService (2).

g You can activate the group status by changing the MIB value indexed by his or her name to one. The value set can be shown in the output of the SNMP operation (**h**).

The following screen shows the updated entries in the SNMP agent configuration file, /etc/snmpd.conf:

```
USM_USER u04 00000002000000000009186895 HMAC-MD5 be2130e2c483d7d2bdf63d14d901a47a  None - L nonVolatile
VACM_GROUP group1 USM u04 nonVolatile                                  i
```

Figure 55. SNMP agent configuration added for the new user

The localized authentication key **i** is generated using the keyChange value set for the usmUserAuthKeyChange object.

If the `osnmp` command is used as an SNMP manager for the newly defined user, you also have to update the SNMP manager's configuration, /etc/osnmp.conf. Follow the steps below:

1. Create the keys from the new password.

   Generate security keys based on the new passwords using the pwtokey utility. In the sample configuration here, we use the authentication key only.

```
KAKKY @ RA03:/u/kakky>pwtokey -p HMAC-MD5 -u auth newauthpass 9.24.104.149
Display of 16 byte HMAC-MD5 authKey:
  ba226e9841527101779f62ca75a8bbf9

Display of 16 byte HMAC-MD5 localized authKey:
  be2130e2c483d7d2bdf63d14d901a47a
```

Figure 56. Generate an authentication key from a password

2. Update /etc/osnmp.conf

   In the SNMP manager's configuration file, you have to define an association between a user and a target host. The following entry is defined for the user u04 and the SNMP agent on RA39.

```
v3m39a2 9.24.104.149 snmpv3 u04 -  - AuthNoPriv HMAC-MD5 ba226e9841527101779f62ca75a8bbf9 - -
```

Then u04 can retrieve the MIB variables from the SNMP agent on R39.

```
KAKKY @ RA03:/u/kakky>snmp -v -h v3m39a2 walk system
sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0
sysObjectID.0 = 1.3.6.1.4.1.2.3.13
sysUpTime.0 = 195800
sysContact.0 = Tatsuhiko Kakimoto
sysName.0 = CS for OS/390 V2R8 on SYSTEM39
sysLocation.0 = ITSO Raleigh, IBM
sysServices.0 = 76
sysORLastChange.0 = 500
sysORID.1 = 1.3.6.1.4.1.2.11.7.1
sysORID.2 = 1.3.6.1.4.1.2.11.7.3
sysORID.3 = 1.3.6.1.4.1.2.11.7.2
sysORDescr.1 = OS/390 SNMP Agent
sysORDescr.2 = OS/390 TCP/IP OSPF SNMP Subagent
sysORDescr.3 = OS/390 TCP/IP SNMP Subagent
sysORUpTime.1 = 0
sysORUpTime.2 = 500
sysORUpTime.3 = 500                                                                    :
```

Figure 57. Retrieve the MIB information using the user u04

3.4 Monitoring performance using MIB information and traps

The SLA Performance Monitor MIB (SLAPM-MIB) subagent provides monitoring through PolicyStats, PolicyMonitor, and Subcomponent tables. The SLA subagent communicates with the SNMP agent to provide GET and SET support for various MIB tables and objects, and also trap generation. Traps can be generated for various performance out-of-bounds conditions, for example, maximum delay exceeded, and for significant events such as policy deletion.

To monitor the SLAP-MIB variables you must get the policy agent up and runing. For the information about the configuration of the policy agent and RSVP agent, refer to Appendix B, "Quality of Service (QoS)" on page 177.

3.4.1 SLAPM-MIB and traps monitored by the SLA subagent

The SLA subagent provides information about service policies and performance data about applications mapped to those policies by using two tables:

Table 17. SLAPM-MIB monitoring tables

SLAPM-MIB Table Name	Description
slapmPolicyStatsTable	Provides information about defined service policies and aggregate performance data for mapped applications.
slapmSubcomponentTable	Provides information about individual TCP or UDP applications and application-specific performance data.

The SLA subagent also supports performance monitoring through the slapmPolicyMonitorTable object. Entries are created in the monitor table to establish the desired criteria for monitoring. The following levels of monitoring are provided:

- Aggregate

  Monitoring is performed based on the aggregate of all TCP or UDP applications that are mapped to one or more service policies.

- Subcomponent

  Monitoring is performed based on a single TCP or UDP application.

Three types of monitoring are provided for measuring application performance:

Table 18. Monitoring types

Monitoring Type	Description
MinRate	The current input/output rates of the application(s) are compared to threshold values established in the monitor table entry. If the current rates are less than the threshold, an SNMP trap is sent if traps are enabled.
MaxRate	The current input/output rates of the application(s) are compared to threshold values established in the monitor table entry. If the current rates are greater than the threshold, an SNMP trap is sent if traps are enabled.
MaxDelay	The current delay rates of the application(s) are calculated by using TCP round trip time (RTT). For aggregate monitoring, the RTT of all TCP applications are averaged. The delay rates are compared to threshold values established in the monitor table entry. If the current rates are greater than the threshold, an SNMP trap is sent if traps are enabled.

Note: MaxDelay monitoring is only available for TCP applications.

You will see the detailed definition of the SLA MIB in the draft RFC, which is shipped with CS for OS/390 IP as /usr/lpp/tcpip/samples/slapm.txt.

The following four sets of SLAPM-MIB objects are supported by the SLA subagent:

- Global objects
- Policy Stats table
- Policy Monitor table

- Subcomponent table

Several MIB objects are used in establishing monitor table entries and for configuring whether and how often traps are sent. Table 19 on page 88 shows significant MIB objects to monitor the performance and/or control the generation of traps.

Table 19. SLAPM-MIB Objects

MIB Object	Action
slapmPolicyMonitorControl	Controls what levels and types of monitoring and/or traps are in effect.
slapmPolicyMonitorInterval	Sets the interval for calculating input/output and delay rates and checking those values against the monitor table thresholds.
slapmPolicyMonitorMinRateLow	Establishes the lower threshold values for MinRate in units of kilobits per second.
slapmPolicyMonitorMinRateHigh	Establishes the upper threshold values for MinRate in units of kilobits per second.
slapmPolicyMonitorMaxRateLow	Establishes the lower threshold values for MaxRate in units of kilobits per second.
slapmPolicyMonitorMaxRateHigh	Establishes the upper threshold values for MaxRate in units of kilobits per second.
slapmPolicyMonitorMaxDelayLow	Establishes the lower threshold values for MaxDelay in units of milliseconds.
slapmPolicyMonitorMaxDelayHigh	Establishes the upper threshold values for MaxDelay in units of milliseconds.
slapmPolicyMonitorRowStatus	The value of this object indicates whether the row is active, in the process of being created, or temporarily not in use.
slapmPolicyTrapEnable	Enables or suppresses the generation of Policy Deleted and Monitor Deleted traps.
slapmPolicyTrapFilter	Establishes the number of times a given MinRate, MaxRate, or MaxDelay event must be encountered before a trap is generated.
slapmPolicyPurgeTime	Specifies a timeout value for Policy Deleted traps.

Note

For CS for OS/390 V2R8, the SLA subagent objects are experimental and occupy the experimental portion of the MIB tree. The MIB objects will change in future releases.

When SNMP traps are enabled, a *not achieved* trap is sent as described in Table 18 on page 87. A corresponding *okay* trap is sent when the traffic once again conforms to the boundaries established in the monitor table entry. See Figure 58 for an overview of performance monitoring.

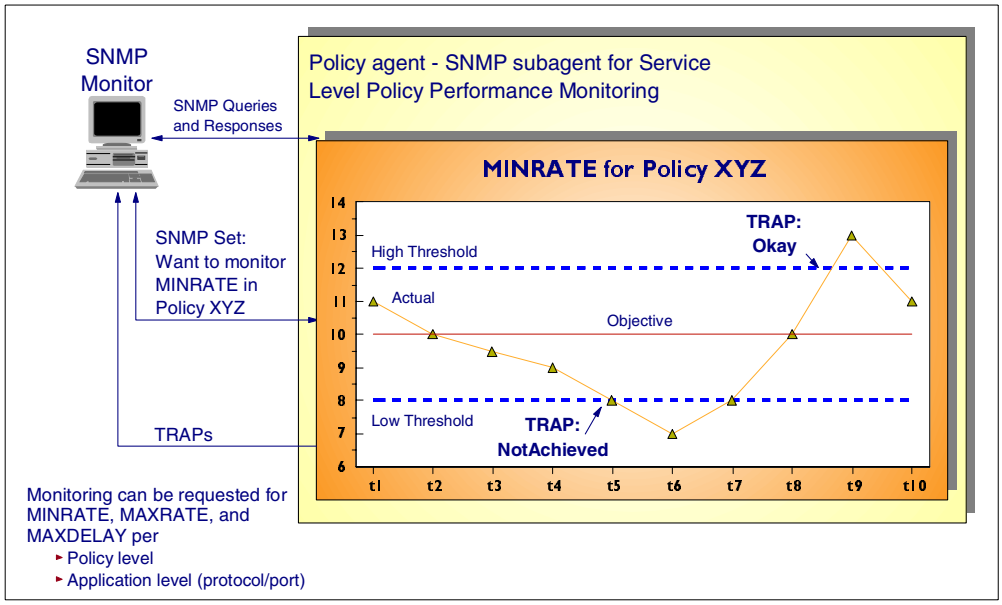


Figure 58. Service level agreement performance monitoring

In addition to the traps used to measure application performance, two additional traps are used to monitor table administration, the PolicyDeleted and MonitorDeleted traps.

The detailed information on the monitoring traps are shown in the table below. These traps are defined as part of the IBM Enterprise-Specific Trap:

Table 20. SNMP TCP/IP subagent enterprise-specific trap types: SLAPM traps

Value	Type	Description
1	slapmMonitoredEventNotAchieved	This notification is generated when a monitored event is not achieved with respect to the threshold. This applies only toward monitoring a policy traffic profile as an aggregate using an associated slapmPolicyStatsEntry. The value of slapmPolicyMonitorControl can be examined to determine what is being monitored. The first slapmPolicyMonitorStatus value supplies the current monitor status while the second value supplies the previous status.

Value	Type	Description
2	slapmMonitoredEventOkay	This notification is generated when a monitored event has improved to an acceptable level. This applies only toward monitoring a policy traffic profile as an aggregate using an associated slapmPolicyStatsEntry. The value of slapmPolicyMonitorControl can be examined to determine what is being monitored. The first slapmPolicyMonitorStatus value supplies the current monitor status while the second value supplies the previous status.
3	slapmPolicyProfileDeleted	A slapmPolicyProfileDeleted notification is sent when a slapmPolicyStatsEntry is deleted if the value of slapPolicyTrapEnable is enabled (1).
4	slapmPolicyMonitorDeleted	A slapmPolicyMonitorDeleted notification is sent when a slapmPolicyMonitor is deleted if the value of slapPolicyTrapEnable is enabled (1).
5	slapmSubcomponentMonitoredEvent NotAchieved	This notification is generated when a monitored value does not achieve a threshold specification. This applies only toward monitoring the individual components of a policy traffic profile. The value of the corresponding slapmPolicyMonitorControl can be examined to determine what is being monitored. The first slapmSubcomponentMonitorStatus value supplies the previous status.
6	slapmSubcomponentMonitoredEvent Okay	This notification is generated when a monitored value has reached an acceptable level.

3.4.2 Extracting the performance information from SLAPM-MIB

To get performance information from SLAPM-MIB, follow the steps below.

Step 1: Start the SNMP agent, the Policy agent and the SLA subagent
If the policy information has been installed in the TCP/IP stack, the SLA subagent creates the following objects automatically. Note that if you do not have the policy information installed, only the Global objects are created.

- Global objects
- Policy status table
- Subcomponent table

At ITSO Raleigh, we used the LDAP server in the OS/390 system to provide the policy information to the Policy agent. Figure 59 shows our policy definition for RA39, whose host name is MVS39A.

```
dn: s=srv39a1, o=IBM_US, c=US
```

```
objectclass: ServiceCategories
selectortag: MVS39A
servicename: srv39a1
tcpimagename: TCPIPA
policyscope: DataTraffic
maxrate: 10000
minrate: 1000
outgoingtos: 10000000

dn: r=rule39a1, o=IBM_US, c=US
objectclass: ServicePolicyRules
selectortag: MVS39A
policyrulesname: rule39a1
tcpimagename: TCPIPA
policyscope: Both
daysofweekmask : 1111111
timeofdayrange :7:00-17:00
direction: Both
permission:Allowed
protocolnumber: TCP
sourceportrange: 12000-12000
servicereference: srv39a1

dn: s=srv39a2, o=IBM_US, c=US
objectclass: ServiceCategories
selectortag: MVS39A
servicename: srv39a2
tcpimagename: TCPIPA
policyscope: DataTraffic
maxrate: 400
minrate: 100
outgoingtos: 11100000

dn: r=rule39a2, o=IBM_US, c=US
objectclass: ServicePolicyRules
selectortag: MVS39A
policyrulesname: rule39a2
tcpimagename: TCPIPA
policyscope: Both
daysofweekmask: 0111111
timeofdayrange: 7:00-20:00
direction: Both
permission:Allowed
protocolnumber: TCP
DestinationPortRange: 20-21
servicereference: srv39a2
```

Figure 59. Policy definitions at ITSO Raleigh

Step 2: Get appropriate index values

To monitor performance with the SLA subagent, the monitor table must be explicitly created using SNMP SET commands. When you create monitor table entries, the appropriate index value has to be specified. The index is composed of the following object values:

- slapmPolicyMonitorOwnerIndex
- slapmPolicyMonitorSystemAddress
- slapmPolicyMonitorPolicyName
- slapmPolicyMonitorTrafficProfileName

See Figure 60 for the structure of the index value:

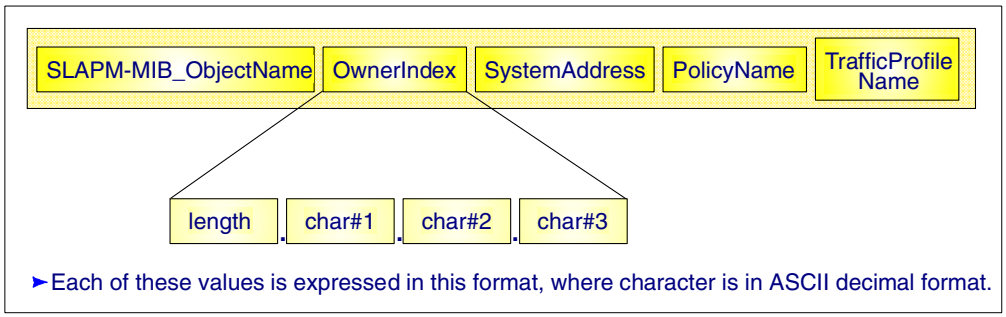


Figure 60. PolicyMonitorTable index structure

For example, the value u1 is expressed as 2.117.49. The PolicyName part of the index matches the value specified on the PolicyName keyword on the ServicePolicyRules statement for the associated policy definition, while the TrafficProfileName matches the rule name specified on the ServicePolicyRules statement. On the other hand, if you use an LDAP server to maintain policy information, the PolicyName matches the value for the PolicyName object, and the PolicyRulesName object is used for the TrafficProfileName part of the index. Figure 61 shows you practical examples of how the index is created.

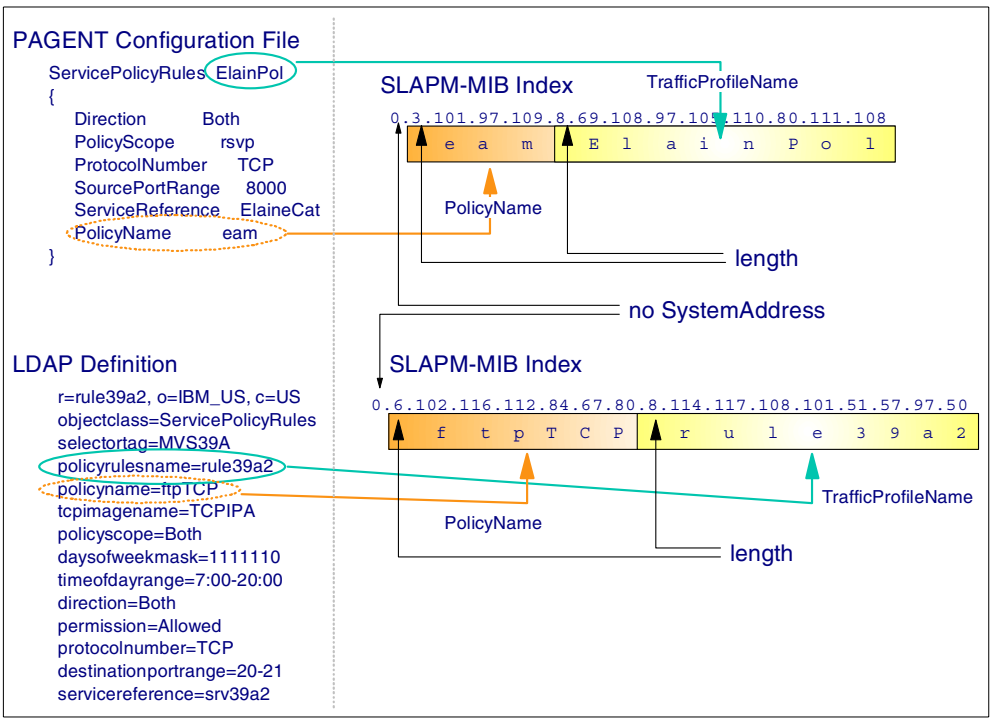


Figure 61. Sample index values

To obtain the index values in your SLAPM-MIB, you can walk through the slapmPolicyStatsTable (see Figure 62). Note that the index value used in the slapmPolicyStatsTable entries consist of the last three values used in the monitor table index, namely the SystemAddress, PolicyName, and TrafficProfileName.

```
KAKKY @ RA03:/u/kakky>osnmp -v -h v3m39a walk slapmPolicyStatsTable
slapmPolicyStatsOperStatus.0.0.8.114.117.108.101.51.57.97.49 = 2
slapmPolicyStatsOperStatus.0.0.8.114.117.108.101.51.57.97.50 = 2
slapmPolicyStatsActiveConns.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsActiveConns.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsTotalConns.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsTotalConns.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsFirstActivated.0.0.8.114.117.108.101.51.57.97.49 = 1999-7-8,8:4:11
slapmPolicyStatsFirstActivated.0.0.8.114.117.108.101.51.57.97.50 = 1999-7-8,8:4:11
slapmPolicyStatsLastMapping.0.0.8.114.117.108.101.51.57.97.49 = 0-0-0,0:0:0.0
slapmPolicyStatsLastMapping.0.0.8.114.117.108.101.51.57.97.50 = 0-0-0,0:0:0.0
slapmPolicyStatsInOctets.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsInOctets.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsOutOctets.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsOutOctets.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsConnectionLimit.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsConnectionLimit.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsCountAccepts.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsCountAccepts.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsCountDenies.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsCountDenies.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsInDiscards.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsInDiscards.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsOutDiscards.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsOutDiscards.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsInPackets.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsInPackets.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsOutPackets.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsOutPackets.0.0.8.114.117.108.101.51.57.97.50 = 0
slapmPolicyStatsMinRate.0.0.8.114.117.108.101.51.57.97.49 = 2400
slapmPolicyStatsMinRate.0.0.8.114.117.108.101.51.57.97.50 = 100
slapmPolicyStatsMaxRate.0.0.8.114.117.108.101.51.57.97.49 = 3200
slapmPolicyStatsMaxRate.0.0.8.114.117.108.101.51.57.97.50 = 400
slapmPolicyStatsMaxDelay.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyStatsMaxDelay.0.0.8.114.117.108.101.51.57.97.50 = 0
```

Figure 62. Walking the slapmPolicyStatsTable

Then, you can cut and paste the index value from the slapmPolicyStatsTable entries, and add an OwnerIndex of your choosing at the beginning of the index. For the above example, the index value from the Policy Stats table is `0.0.8.114.117.108.101.51.57.97.49` for the policy rule rule39a1 and `0.0.8.114.117.108.101.51.57.97.50` for the policy rule rule39a2 (see our policy definitions in Figure 59 on page 91).

Since we chose the owner index of u1, the complete index is:

`2.117.49.0.0.8.114.117.108.101.51.57.97.49` for rule39a1, and

`2.117.49.0.0.8.114.117.108.101.51.57.97.50` for rule39a2.

Step 3: Create the monitor table objects

Now you are ready to create the monitor table objects. The table can be created by issuing several SET commands as shown in Figure 63 on page 94.

```
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyTrapEnable.0 1   1
1.3.6.1.3.88.1.1.7.0 = 1
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyTrapFilter.0 1   2
1.3.6.1.3.88.1.1.8.0 = 1
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyPurgeTime.0 60   3
1.3.6.1.3.88.1.1.6.0 = 60
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorControl.2.117.49.0.0.8.114.117.108.101.51.57.97.49 \'0000003F\'h
1.3.6.1.3.88.1.2.2.1.5.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = '0000003f'h   4
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorMinRateLow.2.117.49.0.0.8.114.117.108.101.51.57.97.49 l
EZZ3322I l is not numeric
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorMinRateHigh.2.117.49.0.0.8.114.117.108.101.51.57.97.49 h
EZZ3322I h is not numeric
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorMaxRateLow.2.117.49.0.0.8.114.117.108.101.51.57.97.49 l
EZZ3322I l is not numeric
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorMaxRateHigh.2.117.49.0.0.8.114.117.108.101.51.57.97.49 h
EZZ3322I h is not numeric
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorMaxDelayLow.2.117.49.0.0.8.114.117.108.101.51.57.97.49 l
EZZ3322I l is not numeric
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorMaxDelayHigh.2.117.49.0.0.8.114.117.108.101.51.57.97.49 h
EZZ3322I h is not numeric
KAKKY @ RA03:/>osnmp -h v3m39a set slapmPolicyMonitorRowStatus.2.117.49.0.0.8.114.117.108.101.51.57.97.49 1
1.3.6.1.3.88.1.2.2.1.23.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 1   5
```

Figure 63. Creating monitor table objects

1 Turn on the generation of Policy Deleted and Monitor Deleted traps.

2 If this value is set to 1, each time a given MinRate, MaxRate or MaxDelay event occurrs, a trap is generated.

3 60 seconds after a service policy is deleted, a Policy Deleted trap is generated.

4 Setting the slapmPolicyMonitorControl to x'3f' enables it to monitor all events and generate all kinds of SLAPM traps. After issuing this command, you will be able to walk through the slapmPolicyMonitorControl table.

5 The slapmPolicyMonitorRowStatus object controls the status of a monitor entry. To activate the entry, set the value to 1.

Note: To change any of the monitor table object values for an existing table entry or row, you must take the row out of service to make the changes. To do this, set the value of slapmPolicyMonitorRowStatus to 2.

Then, you can walk through the monitor table entries:

```
KAKKY @ RA03:/u/kakky>osnmp -v -h v3m39a walk slapmPolicyMonitorTable
slapmPolicyMonitorControl.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = '0000003f'h
slapmPolicyMonitorStatus.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = '00000000'h
slapmPolicyMonitorInterval.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 20
slapmPolicyMonitorIntTime.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 1999-7-8,12:8:2.8
slapmPolicyMonitorCurrentInRate.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorCurrentOutRate.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMinRateLow.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 2159
slapmPolicyMonitorMinRateHigh.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 2640
slapmPolicyMonitorMaxRateHigh.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 3520
slapmPolicyMonitorMaxRateLow.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 2879
slapmPolicyMonitorMaxDelayHigh.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMaxDelayLow.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMinInRateNotAchieves.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMaxInRateExceeds.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMaxDelayExceeds.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMinOutRateNotAchieves.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorMaxOutRateExceeds.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorCurrentDelayRate.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 0
slapmPolicyMonitorRowStatus.2.117.49.0.0.8.114.117.108.101.51.57.97.49 = 1
```

Figure 64. Objects in monitor table

Note that every time you start the SLA subagent, the monitor table has to be created by issuing the commands shown in Figure 64 on page 94. For easy operation, we use the following shell program to initialize and create the necessary table entries:

```
#!/bin/sh

DEST=$1
TMPFILE=./"tmp.""$$"
OWNER_INDEX="2.117.49"   # Owner index is set to "u1"

if [ "$DEST" = "" ]
then
        echo " Usage : $0 dest_SNMP_AgentID "
        exit 1
fi

#   Disable AuthenticationFailure Traps
SNMP_CMD="osnmp -h $DEST set snmpEnableAuthenTraps.0 2"
echo $SNMP_CMD
$SNMP_CMD
if [ $? != 0 ]
then
   exit 255
fi

#   Enable PolicyDeleted and MonitorDeleted Traps
SNMP_CMD="osnmp -h $DEST set slapmPolicyTrapEnable.0 1"
echo $SNMP_CMD
$SNMP_CMD
if [ $? != 0 ]
then
   exit 255
fi

SNMP_CMD="osnmp -h $DEST set slapmPolicyTrapFilter.0 1"
echo $SNMP_CMD
$SNMP_CMD
if [ $? != 0 ]
then
   exit 255
fi

SNMP_CMD="osnmp -h $DEST set slapmPolicyPurgeTime.0 60"
echo $SNMP_CMD
$SNMP_CMD
if [ $? != 0 ]
then
   exit 255
fi

#   Retrieve part of the index from PolicyStatsTable
SNMP_CMD="osnmp -v -h $DEST walk slapmPolicyStatsTable"
$SNMP_CMD | grep slapmPolicyStatsOperStatus | cut -c28-999 > $TMPFILE
if [ $? != 0 ]
then
   exit 255
fi

#   While loop to support multiple policies installed.
cat $TMPFILE |
while read line
do
     set $line
     INDEX="$OWNER_INDEX.$1"
     echo $INDEX

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorControl.$INDEX '0000003F'h"
     echo $SNMP_CMD
     $SNMP_CMD
     if [ $? != 0 ]
     then
        exit 255
     fi

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorMinRateLow.$INDEX 1"
     echo $SNMP_CMD
```

```
     $SNMP_CMD

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorMinRateHigh.$INDEX h"
     echo $SNMP_CMD
     $SNMP_CMD

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorMaxRateLow.$INDEX l"
     echo $SNMP_CMD
     $SNMP_CMD

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorMaxRateHigh.$INDEX h"
     echo $SNMP_CMD
     $SNMP_CMD

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorMaxDelayLow.$INDEX l"
     echo $SNMP_CMD
     $SNMP_CMD

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorMaxDelayHigh.$INDEX h"
     echo $SNMP_CMD
     $SNMP_CMD

     SNMP_CMD="osnmp -h $DEST set slapmPolicyMonitorRowStatus.$INDEX 1"
     echo $SNMP_CMD
     $SNMP_CMD
     if [ $? != 0 ]
     then
        exit 255
     fi
done

#   Display the PolicyMonitorTable objects.
echo ""
SNMP_CMD="osnmp -v -h $DEST walk slapmPolicyMonitorTable"
echo $SNMP_CMD
$SNMP_CMD

/bin/rm $TMPFILE
echo "Table initialization completed."
exit 0
```

Figure 65. Monitor table initialization shell program: MonitorControl.sh

This shell program requires one parameter, the winSNMPName parameter configured in your /etc/osnmp.conf file. Using this shell program, you will create and initialize each monitor table for all policies installed in the TCP/IP stack. The MIB setting could be changed to meet your monitor policy.

Two of the monitor table objects, fields MonitorControl and MonitorStatus, are important in setting up the entries and understanding why traps are generated. Both of these fields have the SNMP data type BITS, which means they are bit strings, where bit 0 is the low order bit. Any combination of bits can be set into these objects. Refer to Table 21 and Table 22 for the possible values:

Table 21. MonitorControl objects values

Object Name	Value (xx54 3210)
0 - monitor MinRate	0000 0001 = x'01'
1 - monitor MaxRate	0000 0010 = x'02'
2 - monitor MaxDelay	0000 0100 = x'04'
3- enable aggregate traps	0000 1000 = x'08'
4 - enable subcomponent traps	0001 0000 = x'10'
5 - monitor subcomponent	0010 0000 = x'20'

Table 22. MonitorStatus objects values

Object Name	Value (xx98 7654 3210)
0 - slaMinInRateNotAchieved	0000 0000 0001 = x'001'
1 - slaMaxInRateExceeded	0000 0000 0010 = x'002'
2 - slaMaxDelayExceeded	0000 0000 0100 = x'004'
3 - slaMinOutRateNotAchieved	0000 0000 1000 = x'008'
4 - slaMaxOutRateExceeded	0000 0001 0000 = x'010'
5 - monitorMinInRateNotAchieved	0000 0010 0000 = x'020'
6 - monitorMaxInRateExceeded	0000 0100 0000 = x'040'
7 - monitorMaxDelayExceeded	0000 1000 0001 = x'080'
8 - monitorMinOutRateNotAchieved	0001 0000 0000 = x'100'
9 - monitorMaxOutRateExceeded	0010 0000 0000 = x'200'

As you can see in Table 22, by extracting the values of the slapmPolicyMonitorStatus objects using the GET SNMP command, you can get the current status of the monitored traffic.

3.4.3 Receiving SLAPM traps

At ITSO Raleigh, we used an application written in MVS C Socket API and FTP clients to monitor the performance status. The user-written application is a client/server application that works as a bulk data transfer. For a customized application, the policy rule rule39a1 is installed, and for FTP clients the rule39a2 is used.

Figure 66 shows the overview of our environment.

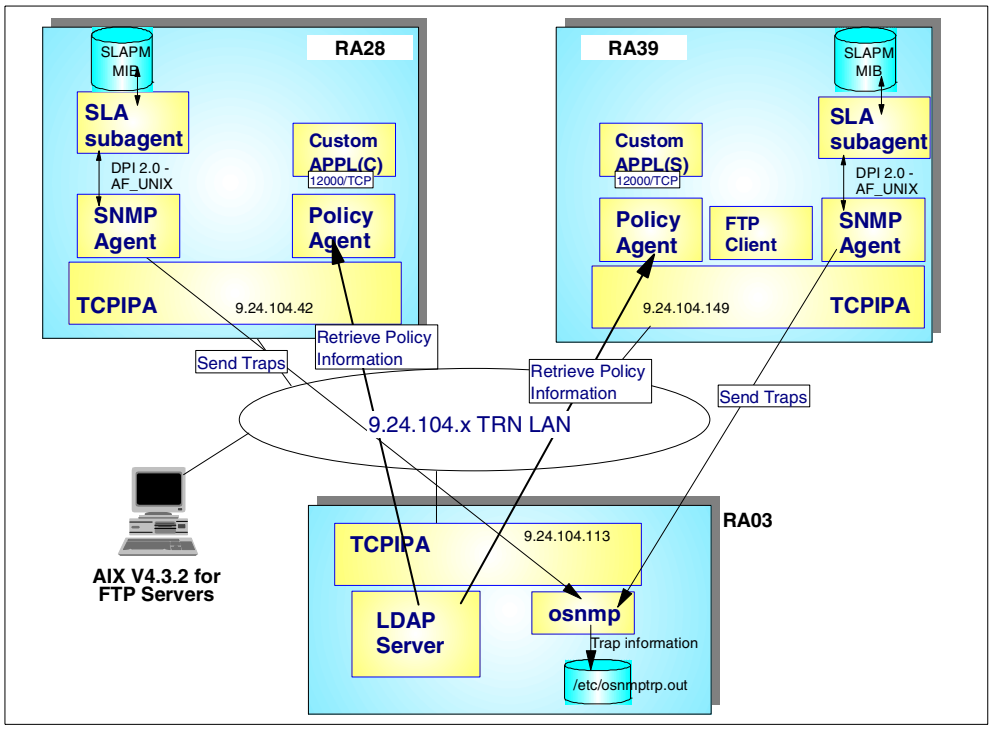


Figure 66. Test environment at ITSO Raleigh

You will see various SLAPM traps on the following pages. For a list of the SLAPM traps, see Table 19 on page 88.

```
Display of SNMPv2 trap with USM security
security-name:   u03
security-level:  AuthNoPriv
security-model:  USM
contextEngineID: '0000000200000000009186895'h
context-name:    NULL
enterprise oid:  1.3.6.1.3.88
       .....     slapmMIB
agent-address:   9.24.104.149 ('09186895'h)  1
generic-trap:    enterpriseSpecific ('00000006'h)
specific-trap    1 ('00000001'h)             1
time-stamp:      817000 - 2 hours, 16 minutes, 10.00 seconds
Trap varBinds:             2
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.8.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorIntTime.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   OCTET_STRING (DateAndTime) 1999-7-4,18:44:39.0    3
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.5.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorControl.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   OCTET_STRING '0000003f'h                          4
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.6.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorStatus.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   OCTET_STRING '00000008'h                          5
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.6.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorStatus.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   OCTET_STRING '00000000'h                          6
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.9.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorCurrentInRate.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   INTEGER 520                                       7
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.10.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorCurrentOutRate.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   INTEGER 521                                       8
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.2.1.22.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        name:    slapmPolicyMonitorCurrentDelayRate.2.117.49.0.0.8.114.117.108.101.51.57.97.49
        value:   INTEGER 12                                        9
```

Figure 67. SLAPM trap: type 1

This is the slapmMonitorEventNotAchieved trap sent from RA39 (1). The index of the object (2) tells you the policy to which this event is corresponding. The

SLAPM traps are defined in the draft RFC, which is shipped with OS/390 V2R8 IP as slapm.txt in the /usr/lpp/tcpip/samples directory, and contains the following objects:

- slapmPolicyMonitorIntTime 3

  This object shows the time stamp when the last interval ended.

- slapmPolicyMonitorControl 4

  The value determines the type and level of monitoring that is applied to this policy.

- slapmPolicyMonitorStatus 5

  The value of the slapmPolicyMonitorStatus object indicates when a monitored value has not met a threshold or is not meeting the defined service level. Most SLAPM traps contain two of these objects. The first object's value shows the current monitor status, and the second one supplies the previous status. The value x'08' means monitorMinOutRateNotAchieved.

- slapmPolicyMonitorStatus 6

  This is the send slapmPolicyMonitorStatus object in this trap, that is, the value describes the previous monitor status. The value x'00' means that in the previous monitor interval, performance met the requirement.

- slapmPolicyMonitorCurrentInRate 7

  The value determines the current in transfer rate in kilobits per seconds.

- slapmPolicyMonitorCurrentOutRate 8

  The value determines the current out transfer rate in kilobits per seconds (kbps). The value 521 kbps is lower than the value of 1000 kbps configured for the MinRate object. For our policy definition refer to Figure 59 on page 91.

- slapmPolicyMonitorCurrentDelayRate 9

  The value for this object shows the current delay rate for this entry in milliseconds. The value is calculated by taking the average of the TCP roundtrip times for all associated slapmSubcomponentTable entries within an interval.

The trap in Figure 68 on page 100 describes the event that has been monitored at an application level.

For SLAPM-MIB objects monitored at an application level, the index consists of the following values:

- slapmSubcomponentRemAddress: the remote address of a subcomponent
- slapmSubcomponentRemPort: the remote port of a subcomponent
- slapmSubcomponentLocalAddress: the local address of a subcomponent
- slapmSubcomponentLocalPort: the local port of a subcomponent

You can determine the application that has encountered the trap events from the index value.

```
Display of SNMPv2 trap with USM security
security-name:   u03
security-level:  AuthNoPriv
security-model:  USM
contextEngineID: '0000000020000000009186895'h
context-name:    NULL
enterprise oid:  1.3.6.1.3.88
       .....     slapmMIB
agent-address:   9.24.104.149 ('09186895'h)     1
generic-trap:    enterpriseSpecific ('00000006'h)
specific-trap    5 ('00000005'h)                2
time-stamp:      829100 - 2 hours, 18 minutes, 11.00 seconds
Trap varBinds:                                                          3
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.6.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentSystemAddress.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    OCTET_STRING                  4
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.7.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentPolicyName.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    OCTET_STRING                  5
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.8.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentTrafficProfileName.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    OCTET_STRING rule39a2         6
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.20.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentMonitorStatus.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    OCTET_STRING '00000200'h      7
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.20.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentMonitorStatus.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    OCTET_STRING '00000000'h      8
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.21.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentMonitorIntTime.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    OCTET_STRING (DateAndTime) 1999-7-4,18:45:27.9      9
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.22.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentMonitorCurrentInRate.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    Gauge32/Unsigned32 0          10
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.23.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentMonitorCurrentOutRate.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    Gauge32/Unsigned32 822        11
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.15.4.9.24.104.241.20.4.172.16.232.39.1049
       name:     slapmSubcomponentTcpRoundTripTime.4.9.24.104.241.20.4.172.16.232.39.1049
       value:    INTEGER 33                    12
```

Figure 68. SLAPM trap: type 5

This trap has been sent from the SNMP agent with engine ID 9.24.104.149 1 which is an IP address configured for the RA39 system in our environment. The trap type is slapmSubcomponentMonitoredEventNotAchieved 2.

The index 3 shows the monitored application is an FTP client that has local IP address 172.16.232.39 and local port number 1049. The monitored connection has been established as an FTP server with IP address 9.24.104.241 on port number 20 (FTP data connection). Note that the IP address 172.16.232.39 is a VIPA address configured on RA39. Since we have the SOURCEVIPA keyword configured in the IPCONFIG statement in PROFILE, the VIPA is used for an FTP client.

The first three objects (4, 5, and 6) describe the global object maintained by the SLA subagent. Since we have configured only the TrafficProfileName object in the policy definition, only this value is shown here.

The current monitor status indicates that the event of monitorMaxOutRateExceeded has occurred 7. The next object shows the previous monitor status 8.

The next three objects (9, 10, and 11) have the same meaning as described in Figure 67 on page 98.

The last object 12 indicates the amount of time that has elapsed, measured in milliseconds, from when the last TCP segment was transmitted by the TCP stack until the ACK was received. When an application is not using a TCP protocol, this value is set to zero.

```
Display of SNMPv2 trap with USM security
security-name:   u03
security-level:  AuthNoPriv
security-model:  USM
contextEngineID: '0000000200000000009186895'h
context-name:    NULL
enterprise oid:  1.3.6.1.3.88
        .....    slapmMIB
agent-address:   9.24.104.149 ('09186895'h)
generic-trap:    enterpriseSpecific ('00000006'h)
specific-trap    6 ('00000006'h)          1
time-stamp:      833200 - 2 hours, 18 minutes, 52.00 seconds
Trap varBinds:
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.6.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentSystemAddress.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   OCTET_STRING
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.7.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentPolicyName.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   OCTET_STRING
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.8.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentTrafficProfileName.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   OCTET_STRING rule39a2
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.20.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentMonitorStatus.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   OCTET_STRING '00000000'h     2
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.20.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentMonitorStatus.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   OCTET_STRING '00000200'h     2
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.21.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentMonitorIntTime.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   OCTET_STRING (DateAndTime) 1999-7-4,18:47:1.1
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.22.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentMonitorCurrentInRate.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   Gauge32/Unsigned32 0
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.23.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentMonitorCurrentOutRate.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   Gauge32/Unsigned32 94
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.3.1.15.4.9.24.104.241.20.4.172.16.232.39.1049
        name:    slapmSubcomponentTcpRoundTripTime.4.9.24.104.241.20.4.172.16.232.39.1049
        value:   INTEGER 39
```

Figure 69. SLAPM trap: type 6

The trap in the screen above is one of the okay traps, slapmSubcomponentMonitoredEventOkay. It is generated when a monitored value has reached an acceptable level.

During our test, this trap was captured next to the trap shown in Figure 68 on page 100. The monitor status indicates the out rate was exceeded by the requested rate in the previous interval (2).

```
Display of SNMPv2 trap with USM security
security-name:   u03
security-level:  AuthNoPriv
security-model:  USM
contextEngineID: '000000020000000000918682a'h
context-name:    NULL
enterprise oid:  1.3.6.1.3.88
       .....     slapmMIB
agent-address:   9.24.104.42 ('0918682a'h)
generic-trap:    enterpriseSpecific ('00000006'h)
specific-trap    3 ('00000003'h)        1
time-stamp:      7321900 - 20 hours, 20 minutes, 19.00 seconds
Trap varBinds:
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.5.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsActiveConns.0.0.8.114.117.108.101.50.56.97.49
        value:   Gauge32/Unsigned32 0
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.6.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsTotalConns.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 7
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.7.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsFirstActivated.0.0.8.114.117.108.101.50.56.97.49
        value:   OCTET_STRING (DateAndTime) 1999-7-4,17:31:45.5
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.8.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsLastMapping.0.0.8.114.117.108.101.50.56.97.49
        value:   OCTET_STRING (DateAndTime) 1999-7-4,18:44:31.9
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.9.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsInOctets.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 73428992
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.10.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsOutOctets.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 73400348
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.11.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsConnectionLimit.0.0.8.114.117.108.101.50.56.97.49
        value:   INTEGER 0
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.12.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsCountAccepts.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 7
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.13.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsCountDenies.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 0
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.14.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsInDiscards.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 0
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.15.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsOutDiscards.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 0
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.16.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsInPackets.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 71308
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.17.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsOutPackets.0.0.8.114.117.108.101.50.56.97.49
        value:   Counter32 70283
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.20.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsMinRate.0.0.8.114.117.108.101.50.56.97.49
        value:   INTEGER 2400
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.21.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsMaxRate.0.0.8.114.117.108.101.50.56.97.49
        value:   INTEGER 3200
varBind oid:     OBJECT_IDENTIFIER 1.3.6.1.3.88.1.2.1.1.22.0.0.8.114.117.108.101.50.56.97.49
        name:    slapmPolicyStatsMaxDelay.0.0.8.114.117.108.101.50.56.97.49
        value:   INTEGER 0
```

Figure 70. SLAPM trap: type 3

This trap is generated when the policy is deleted from a TCP/IP stack, the slapmPolicyProfileDeleted notification 1. When the SLA agent detects the current time is out of the services time defined in the ServicesCategories object, the SLA agent deletes the corresponding slapmPolicyStatsEntry. And then, if the value of slapmPolicyTrapEnable is enabled, this trap is generated.

Following this trap, the slapmPolicyMonitorDeleted trap (type 4) would be generated in most cases.

For more information on the SLAPM-MIB object definition, refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513, *OS/390 SecureWay Communications Server IP User's Guide*, GC31-8514 and the draft RFC.

Chapter 4. IP Network Management with Tivoli NetView for OS/390

As the industry standard for systems and network management, Tivoli NetView for OS/390 offers the most powerful and comprehensive tools and services available for the OS/390 and distributed environments.

Key features that support IP network management include:

- TCP/IP resource monitoring, session management, and automated recovery
- Graphical topology, status monitoring, and command and control from the platform independent NetView Management Console
- IP discovery that uses ICMP and SNMP to discover IP network topology
- Web browser access to Tivoli NetView for OS/390 information
- Enhanced Tivoli NetView for OS/390 3270 management console (Java based)

4.1 Tivoli NetView for OS/390 overview

Tivoli NetView for OS/390 improves your network efficiency, increases system availability, and decreases the overall cost of network and systems management. With NetView, one operator can monitor all of your network and system resources from a single display. With NetView, you can automate many network and systems operations tasks improving the reliability and effectiveness of the systems and networks throughout your business.

Tivoli NetView for OS/390 has the following capabilities:

End-to-end network management - NetView gives you comprehensive end-to-end systems and network management for a variety of resources including SNA, TCP/IP, multiplatform, and multivendor networks.

Automated availability and recovery - With NetView, you can greatly improve your system and network availability using automation routines to monitor your important resources and take actions to recover them in case of failure. You can also reduce the number of routine or repetitive tasks that your operators must perform freeing them to concentrate on more difficult tasks.

Graphical topology with automatic status updates, commands and control - The NetView Management Console graphically displays the current status of your entire network regardless of topology, protocol, or platform and provides point-and-click access to commands and actions to diagnose and correct any problems not handled by your automation routines.

Secure systems and network operation management - A wide range of NetView operator services, from routine operator or timer-driven operations to detailed problem diagnosis and resource recovery, can assist you in operating and managing your systems and networks. NetView also provides extensive function for operating and accessing remote systems. Controlled access to important resources and graphical views protect your critical assets from unauthorized personnel, and extensive command authorization checking protects your systems from accidental or malicious execution of sensitive commands.

Event Consolidation and Management - With NetView, you can receive and process events of almost any type from anywhere in your enterprise. Alerts from hardware devices in your network or from software applications, SNMP traps, Tivoli Enterprise Console events, multi or single-line messages in various formats from applications, and subsystems within your enterprise can all be filtered, logged, processed by automation routines or operator action, or forwarded to another NetView or event management product.

4.1.1 Tivoli NetView for OS/390 IP network management components

Tivoli NetView for OS/390 consists of a comprehensive set of tools that enable you to manage a network from a single point-of-control. The following components can be used to support IP network management:

4.1.1.1 NetView subsystem

Access to the NetView subsystem interface (SSI) is provided in a separate NetView subsystem address space. It enables MVS operators to send commands to NetView and NetView operators to receive output from commands sent to MVS through the SSI. It also enables you to monitor MVS operations because unsolicited messages are sent to NetView through the SSI. However, in most implementations, the NetView program uses extended multiple console support (MCS) for MVS messages and the SSI for the command interface.

The NetView subsystem address space also provides the program-to-program (PPI) interface that enables application programs to send alerts to the NetView program and enables application programs to send data buffers to, or receive data buffers from, other application programs that are running in the same host as the NetView program. Some examples that use the PPI are RODM and GMFHS.

4.1.1.2 Automated Operations Network (AON)

The AON component provides drop-in automation for VTAM SNA, token-ring LAN, and TCP/IP resources using NetView AON automation policy. AON/TCP provides various panels on which you can manage resources in IP network. The supported operations are:

- Issue operator command including TSO, UNIX, and SNMP line mode commands. All of the SNMP operations supported can be done from the AON panels.
- Proactively monitor critical IP resources, such as TCP/IP stacks and application servers. The MIB polling operation with threshold values can be used to monitor resource status.
- TCP/IP session management. Any socket connection can be managed from the AON/TCP panels. If the socket ID goes down, automation can attempt to restart its associated application.

To issue OS/390 UNIX commands from NetView panels, a UNIX command receiver, namely the UNIX for OS/390 command server, has to be started. This server is a dedicated PPI receiver and receives commands and data from NetView. The server process is called CNMEUNIX and runs as a UNIX for OS/390 kernel process.

4.1.1.3 Resource Object Data Manager (RODM)

RODM is an object-oriented data cache that provides a central location for storing, retrieving, and managing operational resource information about any

hardware or software resource in your enterprise. This data store runs in a separate address space and can efficiently handle up to 500,0000 objects.

4.1.1.4 MultiSystem Manager (MSM)

MultiSystem Manager provides an integrated centralized, network management facility that enables you to manage your TCP/IP and other non-SNA network resources from a single workstation. MSM IP agents provide the primary method of discovering IP resources in the OS/390 and distributed environments.

These resources are displayed graphically in NMC or NGMF and actions to control these resources can be initiated by clicking on actions from a pull-down resource-specific action list. Some of the currently supported non-SNA network elements are:

- TCP/IP resources, via agents on Tivoli NetView on AIX, NT, Sun Solaris, HP OpenView on HP-UX or Sun Solaris
- Novell NetWare resources using the agent for NetWare
- Token-ring LAN environment using the LAN Network Manager
- Tivoli resources and distributed monitors using an agent on a Tivoli Management Region
- ATM
- NetFinity

MultiSystem Manager is required to implement the IP discovery function.

4.1.1.5 Tivoli NetView IP discovery

IP discovery uses ICMP and SNMP to discover IP network topology. ICMP ping is used to detect if an interface is up or down and SNMP is used to query the system information. System information includes how many interfaces are in the system, whether the system is a router or bridge and the network which the system is in along with which router each interface is connected to. Select information is sent to NetView over the NetView PPI where it is reformatted and loaded into RODM.

4.1.1.6 NetView management console (NMC)

The NetView management console (NMC) of Tivoli NetView for OS/390 V1R3 graphically displays the resources that represent a network, a portion of a network, or a group of networks at various levels of detail. These views show the network and systems resources that you are monitoring. When you monitor a network, resource status changes are reflected graphically in the views.

NMC consists of a server and a Java-based console, which are generically referred to in this book and the online help as the topology server and topology console. The topology server interacts with GMFHS and RODM and provides information for display on the topology console.

4.1.1.7 Java SNMP services

The NMC MIB browser and Java SNMP command require Java SNMP services on OS/390 UNIX. Before you enable the MIB browser or the Java SNMP commands, you need to start Java SNMP services, either through NetView or directly from OS/390 UNIX System Services.

If you do not plan to use the NMC MIB browser, and you do not need the Java SNMP JTRAP command, you may prefer to use the `osnmp` command shipped with CS for OS/390 IP, instead of the NetView Java snmp command.

Figure 71 illustrates the overview of the IP network management components supported in Tivoli NetView for OS/390 V1R3.

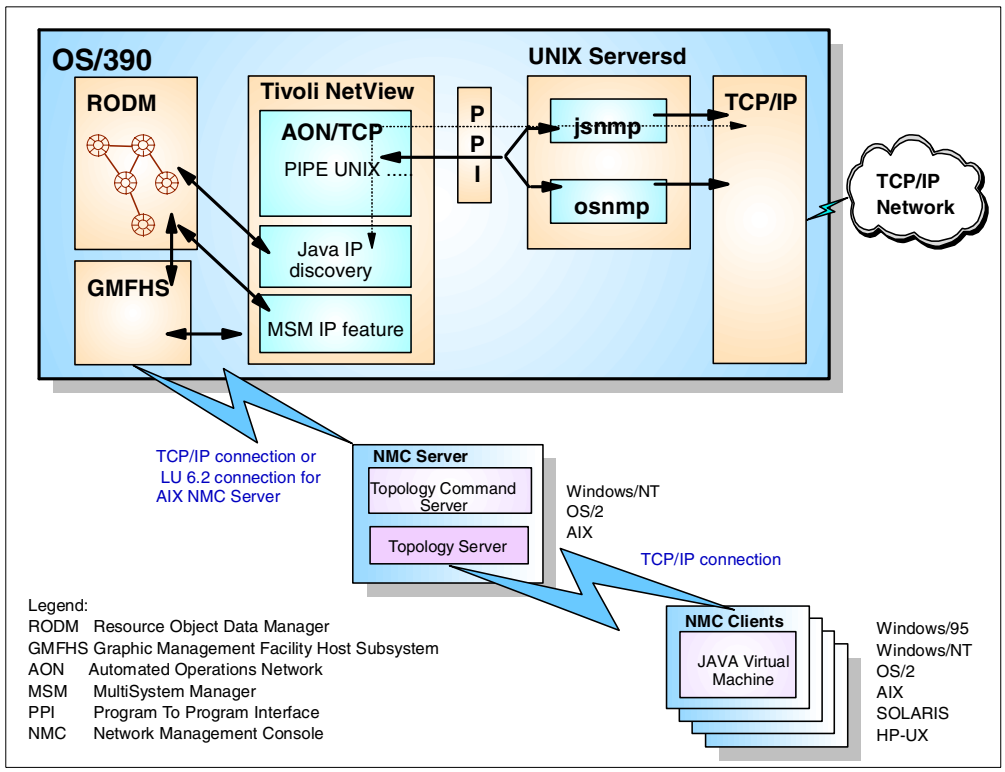


Figure 71. Tivoli NetView for OS/390 IP network management components

Tivoli NetView for OS/390 has two platform-dependent interfaces:

- 3270 interface is the mainframe console and the original NetView interface. It is a text-based interface running on a 3270 display.
- NetView Graphical Management Facility (NGMF) is an OS/2-based graphical interface. NGMF displays topology and status information about network and system resources in a color-coded graphical format.

  Note: Tivoli NetView for OS/390 V1R3 will be the last release to support this function.

Tivoli NetView for OS/390 has three platform independent interfaces:

- NetView Management Console (NMC) is a Java-based graphical user interface that gives you the freedom to monitor and manage the resources that make up your entire enterprise from the platform of your choice: Windows 95, Windows 98, Windows NT, OS/2, or UNIX. NMC is supported on platforms that support Java Version 1.1.6 or above.
- NetView 3270 Console is a Java interface that gives you NetView access from Web browsers, new client platforms, and most Java-enabled platforms. The client provides an enhanced graphical user interface to simulate 3270 functions for Tivoli NetView for OS/390 using TCP/IP connectivity.

- NetView from a Web browser - With Tivoli NetView for OS/390 Web server enhancement, operators can now access NetView and issue commands through a Web browser. This interface can be run from any Web browser on any platform.

4.2 Tivoli NetView for OS/390 configuration

To install the NetView products in our environment we just followed the standard procedure described in the *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236.

The following are the basic steps to implement this function:

1. Install the NetView product. Please refer to Appendix D, “NetView sample configuration” on page 241 for more details.
2. Install the TCP/IP feature of AOC (AON/TCP). AON is shipped within NetView and can be installed with three different automation options: SNA, LAN and TCP/IP depending on customer requirements. We installed AON/TCP only in our environment. Please refer to Appendix D.2, “NetView AON/TCP installation” on page 244 for more details.
3. Install MultiSystem Manager (MSM). Please refer to Appendix D.3, “Installing MultiSystem manager” on page 249 for more details.
4. Install UNIX server. Please refer to Appendix D.4, “UNIX command server installation” on page 253 for more details.
5. Install the Java SNMP services. Please refer to Appendix D.5, “Install the Java SNMP services” on page 255 for more details.

Note

If you do not plan to use the NMC MIB browser, and you do not need the Java SNMP JTRAP command, then the Java SNMP services is not required.

6. Install the Topology Server, the Topology Client and the Specific Resource Command Set on one or more workstations. Please refer to Chapter 5, “NetView Management Console (NMC)” on page 143 for further help.

This is a descriptive table of all the components available with NetView and the explanation of their function:

Figure 72. Tivoli NetView components

Name	Function	Real Started Task Name	Needed for SNMP management only	Needed for NMC
NetView subystem	Allows communication between NetView and OS/390, and NetView programs	NETVS03	YES	YES
NetView procedure	NetView base product	NETVA03	YES	YES
UNIX server	Provide PPI functions from NetView toward OS/390 UNIX	CNMEUNIX	YES	YES
RODM	Data cache for network objects	EKGXRODM	NO	YES

Name	Function	Real Started Task Name	Needed for SNMP management only	Needed for NMC
GMFHS	Builds view based on RODM objects	CNMGMFHS	NO	YES

4.3 Running the SNMP manager from Tivoli NetView for OS/390

In general, any TCP/IP, SNMP or UNIX command can be issued from NetView, including SNMP GET, GETNEXT, SET and WALK, in addition to PING, TRACERTE and NETSTAT. These commands can also be used in REXX CLISTs and other automation routines.

4.3.1 NVSNMP command: command line interface

A new command, NVSNMP, is available to issue SNMP requests. You can use NVSNMP to access a full-screen 3270 panel interface or issue the request for a correlated response.

To issue SNMP commands from NetView using the NVSNMP command, one of the following is required:

1. The `osnmp` command that comes with CS for OS/390 V2R5 IP or later
2. The Tivoli Java SNMP service (the `jsnmp` command) supplied with Tivoli NetView for OS/390 V1R3, that requires:
   - Java for OS/390 at the Java Development Kit (JDK) 1.1.6 level or later
   - IBM TCP/IP Version 3 Release 2 for MVS/ESA (5655-HAL) or OS/390 Version 2 Release 5 (5647-A01) IP Services or later

The NVSNMP command can be issued from either the TCP/AON full screen panels or NetView command line. If the NVSNMP command is issued with no arguments then you get the full screen interface in which all parameters can be specified. If it is issued with one or more arguments then you get the command line interface (REXX API) and all arguments must then be specified.

Figure 73 shows the NVSNMP command syntax.

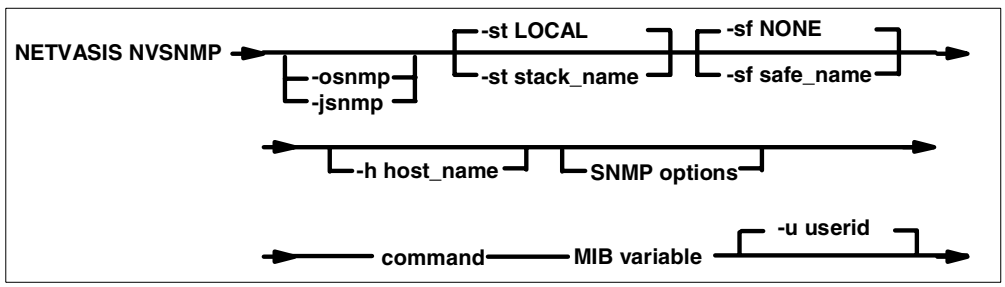


Figure 73. NVSNMP Syntax

Note: NETVASIS is required to keep command and parameters mixed case.

You can use the Tivoli Java SNMP manager, the `jsnmp` command, or CS for OS/390 IP SNMP manager, the `osnmp` command, depending on default settings. The defaults can be set through the AON policy statement in member

DSIPARM(FKXCFG01) when defining your MVS stack with the TCP390 statement (SNMP=MVS | JAVA) or can be defined as a system-wide default with the TCP390 DEFAULTS statement. Please refer to Table 27 on page 242 for the data set definitions in our test environment.

If you are not running AON and want to use Tivoli Java SNMP, set the FKXSP.stack_name.SNMP common global variable in an initialization command list, where stack_name is the TCP/IP service point name. If this variable is not set, the CS for OS/390 IP SNMP manager will be used, which is default.

In our implementation, we specified these variables in a REXX CLIST using the following REXX statements:

```
/* REXX */
FKXSP.NMPNET03.SNMP = 'jsnmp'
'GLOBALV PUTC FKXSP.NMPNET03.SNMP'
```

Note: NetView global variables are used to store environment information. There are two types of global variables: Task and Common. Task global variables are accessible only to the NetView task under which they were defined or set. Common global variables are accessible to any NetView task. Please refer to *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236 and *Tivoli NetView for OS/390 Automation Guide Version 1 Release 3*, SC31-8225 for further information.

The NVSNMP command accepts the following operands:

- -osnmp: Specifies to issue the request to TCP/IP for MVS SNMP. This is the default if FKXSP.NMPNET03.SNMP = 'jsnmp' is not specified in the AON policy statement.
- -jsnmp: Specifies to issue the request to Java SNMP. If not specified, the FKXSP.*.SNMP common global variable(s) are checked to determine which agent to use.
- -st stack_name: Specifies the TCP/IP service point to which the command is sent. The default is LOCAL (MVS only).
- -sf safe_name: Specifies an input safe which contains MIB variable names to be passed to the UNIX pipeline. The default is NONE (MVS only).
- -h hostname: Specifies the host name or TCP/IP address of the host from where the SNMP MIBs are to be pulled.
- SNMP_options: Specifies valid SNMP options.
- command: Specifies the name of the command. Valid names are as follows:

  ```
  GET | GETBULK
  GETNEXT
  WALK | BULKWALK
  SET
  JTRAP (Java only)
  ```

- MIB_variable: Specifies the variable name to be acted upon by the command. If a safe name is used, the MIBs should not be listed on the command line.
- -u userid: Specifies the user ID issuing the Java SNMP request.

If your NVSNMP commands are timing out very often, tune the OS/390 UNIX system for Java. Refer to the website:

```
http://www.s390.ibm.com/products/oe/bpxa1tun.html
```

You may also have to increase the timeout value set for this command by setting global variable `SNMPTIME` which defaults to 29 seconds, for example:

```
SNMPTIME=99
```

In the remaining part of this section, we will show several examples of the NVSNMP usage.

Example 1 - GET command

In the following example we issued the SNMP GET command to query the value of MIB variable `sysContact.0` 1

```
NCCF                    Tivoli NetView   RA03N LUCY     03/01/00 17:59:52
| RA03N
START OF DATA
1.3.6.1.2.1.1.4.0 = Tatsuhiko Kakimoto
END OF DATA




???
netvasis nvsnmp -st LOCAL -h 9.24.104.113 -c publicv1 get
sysContact.0  1
```

Figure 74. NVSNMP get output

Example 2 - WALK command

In this example we issued the SNMP WALK command for the MIB variable `system` 2.

```
NCCF                    Tivoli NetView   RA03N LUCY     03/01/00 19:08:36
| RA03N
START OF DATA
1.3.6.1.2.1.1.1.0 = SNMPv3 agent version 1.0 with DPI version 2.0
1.3.6.1.2.1.1.2.0 = 1.3.6.1.4.1.2.3.13
1.3.6.1.2.1.1.3.0 = 215000
1.3.6.1.2.1.1.4.0 = Tatsuhiko Kakimoto
1.3.6.1.2.1.1.5.0 = CS/390 V2R8 TCP/IP on MVS03
1.3.6.1.2.1.1.6.0 = ITSO Raleigh
1.3.6.1.2.1.1.7.0 = 0
1.3.6.1.2.1.1.8.0 = 2500
1.3.6.1.2.1.1.9.1.2.1 = 1.3.6.1.4.1.2.11.7.1
1.3.6.1.2.1.1.9.1.2.2 = 1.3.6.1.4.1.2.11.7.2
1.3.6.1.2.1.1.9.1.2.3 = 1.3.6.1.4.1.2.11.7.3
1.3.6.1.2.1.1.9.1.3.1 = OS/390 SNMP Agent
1.3.6.1.2.1.1.9.1.3.2 = OS/390 TCP/IP SNMP Subagent
1.3.6.1.2.1.1.9.1.3.3 = OS/390 TCP/IP OSPF SNMP Subagent
1.3.6.1.2.1.1.9.1.4.1 = 0
1.3.6.1.2.1.1.9.1.4.2 = 2500
1.3.6.1.2.1.1.9.1.4.3 = 2500
END OF DATA
???
netvasis nvsnmp -st LOCAL -h v1 -c publicv1 walk system 2
```

Figure 75. NVSNMP WALK system output

Example 3 - SET command
In this example we used the SNMP SET command 3 to change the value of MIB variable `sysContact.0` 5. The previous value 4 was shown with the SNMP GET command.

```
NCCF                      Tivoli NetView   RA03N LUCY     03/01/00 17:57:17
| RA03N
START OF DATA
1.3.6.1.2.1.1.4.0 = Tatsuhiko Kakimoto     4
END OF DATA
| RA03N
START OF DATA
1.3.6.1.2.1.1.4.0 = Lucia Santucci         5
END OF DATA


???                                                          3
netvasis nvsnmp -st LOCAL -h 9.24.104.113 -c publicv1 set
sysContact.0 "Lucia Santucci"
```

Figure 76. NVSNMP SET output

4.3.2 NVSNMP command from a CLIST

It is also possible to issue the `NVSNMP` command in REXX CLISTs using the NetView `PIPE` command. In this way, you can parse, manipulate and format the output of any of these commands using the REXX language. The support is also extended to all other IP, SNMP and UNIX commands available with Tivoli NetView for OS/390.

The following are some examples of REXX CLISTs implementing this new feature.

Example 1 - Issue NVSNMP from a CLIST
This simple REXX CLIST issues a `PIPE` command using the NETV stem to send an SNMP WALK command and displays the results on the NCCF console.

```
****************************** Top of Data ******************************
/* WALKSYS - Sample CLIST to issue an SNMP system walk for an         */
/*           MVS IP stack.                                            */
trace err
address NETVASIS
"pipe netv "||,
     "nvsnmp -v -c publicv1 -h v1 walk system" ||,
     "| stem lucy."||,
     "| cons"
/*                                                                   */
/* insert cust code here to parse/manipulate/etc.....                */
/*                                                                   */
***************************** Bottom of Data ****************************
```

Figure 77. WALKSYS sample CLIST DSICLDU(WALKSYS)

And here is the output:

```
 NCCF                      Tivoli NetView   RA03N LUCY     03/01/00 19:16:28
 | RA03N
 START OF DATA
 sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0
 sysObjectID.0 = 1.3.6.1.4.1.2.3.13
 sysUpTime.0 = 260000
 sysContact.0 = Tatsuhiko Kakimoto
 sysName.0 = CS/390 V2R8 TCP/IP on MVS03
 sysLocation.0 = ITSO Raleigh
 sysServices.0 = 0
 sysORLastChange.0 = 2500
 sysORID.1 = 1.3.6.1.4.1.2.11.7.1
 sysORID.2 = 1.3.6.1.4.1.2.11.7.2
 sysORID.3 = 1.3.6.1.4.1.2.11.7.3
 sysORDescr.1 = OS/390 SNMP Agent
 sysORDescr.2 = OS/390 TCP/IP SNMP Subagent
 sysORDescr.3 = OS/390 TCP/IP OSPF SNMP Subagent
 sysORUpTime.1 = 0
 sysORUpTime.2 = 2500
 sysORUpTime.3 = 2500
 END OF DATA
 ???
WALKSYS
```

Figure 78. WALKSYS CLIST output

Example 2 - PING thresholding

The following CLIST receives three input parameters: a resource name, a count and a threshold value, and issues the `oping` command using these parameters. The resulting output lines are stored in a stem variable and then sent to the NCCF console, but only after the threshold value is checked.

```
***************************** Top of Data **********************
/* TSTPINGC - Sample CLIST to ping a resource, compare results to     */
/*            predefined criteria and display pings whose response    */
/*            time exceeds the criteria.                              */
trace err
parse arg resname count thresh
address NETVASIS
if count = '' then count = 10
if thresh = '' then thresh = .010
"pipe netv "||,
     "IPCMD cmd=oping -c "count" -t 5 "resname ||,
     "| stem lucy."
do i = 1 to lucy.0
  parse var lucy.i . 'took' pingsec 'seconds.'
  if strip(pingsec) > thresh then
     say lucy.i
end
****************************BottomofData***************************
```

Figure 79. TSTPINGC sample CLIST DSICLDU(TSTPINGC)

And this is the output:

```
NCCF                    Tivoli NetView   RA03N LUCY     03/01/00 11:18:57
* RA03N   TSTPINGC 9.24.106.77 20 .008
C RA03N    Ping #1 response took 0.009 seconds.
C RA03N    Ping #2 response took 0.009 seconds.
C RA03N    Ping #4 response took 0.009 seconds.
C RA03N    Ping #7 response took 0.009 seconds.
C RA03N    Ping #9 response took 0.009 seconds.
C RA03N    Ping #10 response took 0.009 seconds.
C RA03N    Ping #11 response took 0.009 seconds.
C RA03N    Ping #12 response took 0.009 seconds.
C RA03N    Ping #13 response took 0.009 seconds.
C RA03N    Ping #16 response took 0.009 seconds.
C RA03N    Ping #17 response took 0.009 seconds.
C RA03N    Ping #18 response took 0.009 seconds.
C RA03N    Ping #20 response took 0.010 seconds.
------------------------------------------------------------------------------


???
```

Figure 80. TSTPINGC output

In the previous example we used the TSTPINGC CLIST to ping host 9.24.106.77 20 times and to show only the response times greater than 0.008 seconds.

4.3.3 NVSNMP command: AON full screen interface

There are two ways to access the SNMP command's full screen interface:

- Walk through the AON menus
- Issue the NVSNMP command with no arguments

In the full screen panel it is possible to have pre-defined logical MIB groups. A user can just tab to the name of the group to retrieve and the entire "group" is retrieved. Also, additional groups can be added.

For CS for OS/390 IP stacks to be managed from the AON panels, you have to define the stack characteristics in an AON control file, such as DSIPARMU(FKXCFG01) for our installation. The following statement defines the IP390 service point for the TCP/IP stack on RA03.

```
TCP390      NMPNET03,             1
            IPADDR=9.24.104.113,  2
            HIER2=SP-APPL,
            HIER3=NETSP,
            DOMAIN=LOCAL,         3
            TCPNAME=TCPIPA,       4
            UNIXSERV=YES,
            FORMAT=STACK,
            SNMP=MVS,             5
            HOSTNAME=mvs03a       6
```

Figure 81. TCP/IP stack definition in the AON control file, FKXCFG01

The following are some attributes that have been defined in this statement:

- The name of the NetView service point 1.
- The IP address 2 and host name 6 of this service point.
- The NetView domain that owns this MVS service point. LOCAL indicates the TCP/IP stack is running on the same host as AON 3.

- The TCP/IP address space name with which AON should have an affinity 4.
- 5 specifies which SNMP agent AON should use when issuing SNMP requests. The supported values are either MVS for the SNMP manager of CS for OS/390 IP (osnmp) or JAVA for Java SNMP services of Tivoli NetView for OS/390 (jsnmp) 5.

For complete information on the AON control file definition, refer to *Tivoli NetView for OS/390 Administration Reference Version 1 Release 3*, SC31-8222.

You will also have to start the UNIX command server before working on the AON panels. See Appendix D, “NetView sample configuration” on page 241 for the detailed information on the NetView configuration.

To open the AON main menu, enter `AON` from an NCCF panel.

```
EZLK0000              AON: Operator Commands Main Menu            RA03N


Select an option

         4 0.   Tutorial
           1.   AON Base Functions
           2.   SNA Automation
           3.   LAN Automation
           4.   TCP/IP Automation




Command ===>
F1=Help      F2=End         F3=Return                              F6=Roll
                                                                 F12=Cancel
```

Figure 82. AON Main Menu

Enter option 4, TCP/IP Automation:

```
FKXK0000         TCP/IP Automation: Commands Menu                 RA03N


Select an option

        2     1.   Netview for AIX Menu
              2.   MVS TCP/IP Menu




Command ===>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                                                                 F12=Cancel
```

Figure 83. TCP/IP Automation: Commands Menu

Enter option 2 for MVS TCP/IP Commands Menu.

```
FKXK2000       MVS TCP/IP Commands Menu                          RA03N


Select an option

         5 1. Issue Ping
           2. MVS IP Session Status
           3. Issue Tracerte Command
           4. Issue Commands
           5. SNMP MENU
           6. IP Server Management
           7. IP Resource Manager



Command ===>
F1=Help      F2=Main Menu   F3=Return                          F6=Roll
                                                              F12=Cancel
```

Figure 84. MVS TCP/IP Commands Menu

From the MVS TCP/IP Commands Menu panel we can issue TCP/IP commands toward our TCP/390 service point. The available options are:

1. Issue Ping: Issue a ping to a machine and display the response. A ping will send an Internet Control Message Protocol (ICMP) echo request to obtain an echoed response. This response is what is displayed.
2. MVS TCP/IP Session Status: Obtain active connection (such as Telnet and FTP) information for an IP address or hostname from one or more service points. Sessions are listed and TCP/IP commands can be issued for the session or service point. You will see further information on this menu in 4.4.1, “Service points management” on page 124.
3. Issue Tracerte: Traceroute sends UDP requests with varying TTL (time to live) values and then waits for the routers between the local and remote hosts to send TTL-exceeded messages.
4. Issue Commands: TSO and UNIX line mode commands can be issued and the response will be windowed back to NetView. TSO stacks will process TSO commands only and UNIX stacks will process UNIX commands only.
5. SNMP Menu: Link to the TCP/IP for MVS SNMP menu. If SNMP support is not installed, this will not be selectable.
6. TCP/IP Server Management: Support for starting and stopping your TCP/IP servers. Additionally, you can see PPI buffer queue statistics.
7. IP Resource Manager: Displays the IP resource manager panel. See 4.4.4, “Proactive monitoring and MIB thresholding” on page 138 for detailed information on this menu.

We selected option 5 for SNMP MENU, which is shown in Figure 85.

```
FKXK2500            TCP/IP for 390 SNMP Menu                    RA03N

 Host Name or IP Address     _____________________________
 (blank: Use Stack Name)     _____________________________

 TCP/IP Stack  ________    (? for Selection list)


    1.Command:

  _ Get    _ GetNext     _ Set    _ Walk


  _ 2.Group Menu


  _ 3.Remote Ping




Command ===>
F1=Help       F2=Main Menu   F3=Return                      F6=Roll
                                                           F12=Cancel
```

Figure 85. TCP/IP for 390 SNMP Menu

From this panel it is possible to issue any supported SNMP command. The following sections provide a brief explanation of each available option.

4.3.3.1 Option 1 - Get, GetNext, Set and Walk

The first available option allows you to issue any SNMP command even if you don't exactly know the command syntax. The following is an example of the SNMP GET command:

Example 1 - GET

From the main panel:

1. Specify the target host **1**.
2. Specify the OS/390 TCP/IP stack or "?" for the list of all your defined stacks **2**. Then press Enter.

```
FKXK2500            TCP/IP for 390 SNMP Menu                    RA03N

 Host Name or IP Address     9.24.106.113 1_________________
 (blank: Use Stack Name)     _______________________________

 TCP/IP Stack  ? 2_____    (? for Selection list)


    1.Command:

  _ Get    _ GetNext    _ Set    _ Walk


  _ 2.Group Menu


  _ 3.Remote Ping




Command ===>
F1=Help      F2=Main Menu   F3=Return                          F6=Roll
                                                              F12=Cancel
```

Figure 86. Example of GET command

3. Select your TCP/IP stack 3 and press Enter.

   This step is optional if the user keys the TCP/IP stack name into the TCP/IP Stack field on panel FKXK2500 as shown in Figure 86.

```
EZLKSLCT          Operator Command Interface: SELECTION          RA03N

Select one of the following.  Then press enter.

                                              UNIXSERV        SYSTEM
3 x NMPNET03                      TCP390      YES


Command ===>
```

Figure 87. Example of GET command

4. Select the Get command 4. Then press Enter.

```
FKXK2500            TCP/IP for 390 SNMP Menu                    RA03N

 Host Name or IP Address     mvs03a_____________________________
 (blank: Use Stack Name)     _____________________________

 TCP/IP Stack  NMPNET03    (? for Selection list)


    1.Command:

4 x Get    _ GetNext    _ Set    _ Walk


  _ 2.Group Menu                          Just press ENTER

  _ 3.Remote Ping


FKX910I SERVICE POINT NAME SET. PRESS ENTER TO CONTINUE
Command ===>
F1=Help      F2=Main Menu   F3=Return                          F6=Roll
                                                              F12=Cancel
```

Figure 88. Example of GET command

5. Insert the community name 5 and specify one or more MIB variables 6. Then press Enter.

```
FKXK2510          TCP/IP for 390 SNMP Commands                RA03N

 Resource: mvs03a
 Password(Community) 5
 TCP/IP Stack: NMPNET03      Command:  GET

MIB Variables:
 sysDescr.0 6_____________________________________________________
 sysUpTime.0_____________________________________________________
 sysContact.0____________________________________________________
 ________________________________________________________________
 ________________________________________________________________
 ________________________________________________________________
 ________________________________________________________________
 ________________________________________________________________
 ________________________________________________________________
 ________________________________________________________________



Command ===>
F1=Help      F2=Main Menu   F3=Return                          F6=Roll
                            F9=Options                        F12=Cancel
```

Figure 89. Example of GET command

6. The output will show up on the next panel (see Figure 90 on page 119).

```
CNMKWIND OUTPUT FROM  SNMP GET to mvs03a                    LINE 0 OF 5
*------------------------------ Top of Data -------------------------------*
START OF DATA
sysDescr.0 = SNMPv3 agent version 1.0 with DPI version 2.0
sysUpTime.0 = 1042600
sysContact.0 = Tatsuhiko Kakimoto
END OF DATA
*------------------------------ Bottom of Data ----------------------------*




TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>
```

Figure 90. Example of GET command

4.3.3.2 Option 2 - Group menu

The second option of the SNMP Menu allows the retrieval of MIB variables logically grouped. We can have three different types of groups: List, Table or Walk.

In the List group we define exact names of MIB variables to be retrieved, for example:

- sysDescr.0
- sysObjectID.0
- sysUpTime.0
- sysContact.0
- sysName.0
- sysLocation.0
- sysServices.0
- sysORLastChange.0

In the Walk group we can have well-known group names (defined in an existing RFC, RFC 1907 for v2 or RFC 1450 for v1) that can be used by the SNMP WALK command, for example:

- system
- snmp
- atTable
- ipNoTab
- icmp

In the Table group, we can use index variables to start the search. There is no need to have variables listed. The table will be walked and all variables in the table will be collected. For table type groups, a starting MIB referring to the table index is needed to walk the table. This is given in ASN.1 notation.
An example could be:

```
INDEX MIB:

    1.3.6.1.2.1.7.5

MIB Variables:

    udpLocalAddress

    udpLocalPorts
```

So, selecting option 2 from the AON SNMP menu, will display the following panel:

```
FKXK2520          TCP/IP for 390 SNMP Groups                    RA03N

                                                          More   :    +
 Resource: 9.24.104.113
Password(Community)
 TCP/IP Stack: NMPNET03

Groups:
system          WALK     ip              WALK    3172sys         WALK
sys2            LIST     ipFoward        WALK    3172ifTrap      WALK
sysOR           TABLE    ipAddrTable     WALK    3172channel     WALK
snmp            WALK     ipRouteTable    WALK    3172lan         WALK
IFTable         WALK     ipNoTab         LIST    3172blk         WALK
IFXTable        WALK     icmp            WALK    3172dblk        WALK
atTable         WALK     tcp             WALK    3172device      WALK
UDP             WALK     tcpConnTable    WALK    MvsTcpSystem    WALK
UDPnotab  1     LIST     tcpNoTab        LIST    MvsTcpIf        WALK
UDPTable        TABLE    atm             WALK    MvsTcpPort      WALK

Command ===>
ENTER=Get Group Data
F1=Help      F2=Main Menu   F3=Return F4=Description          F6=Roll
F7=Backward  F8=Forward     F9=Options                      F12=Cancel
```

Figure 91. SNMP Groups main menu

From this panel, you can see a list of all the defined groups. Group definitions can be found in member DSIPARMU(FKXSNMP) and can be modified. Also, user groups can be added if needed.

The cursor can be placed over the name of a group and pressing the PF4 key will show more details about that particular group.

For example, put the cursor on the UDPnotab group line and press PF4 1.

The following panel is the result of pressing the PF4 key and shows a detailed description of the UDPnotab list group.

```
FKXK2524        TCP/IP for 390 SNMP Groups Description          RA03N

 Group: UDPnotab                Type: LIST

Abstract:
  UDP group MIBs without the UDP Table


 MIB Variables:
  udpInDatagrams.0
  udpNoPorts.0
  udpInErrors.0
  udpOutDatagrams.0




Command ===>
F1=Help      F2=Main Menu   F3=Return                              F6=Roll
                            F9=Options      ENTER=Get Group Data    F12=Cancel
```

Figure 92. more detail of a group

4.3.3.3 Option 3 - Remote Ping

Remote Ping allows you to have a remote host (not the TCP/IP stack) send a Ping message to another host. This helps in problem resolution by providing more flexible connectivity testing. In order to perform the remote ping function, the *from* host must be running OS/390 V2R5 or higher to support the *ibmMvsRPingResponseTime* MIB variable.

Remote ping is `osnmp` only, so the used hostnames must match the ones in the /etc/osnmp.conf file (see Appendix E.4, “/etc/osnmp.conf” on page 283).

To use remote ping, enter the name of the resource which will issue the command **1**, select option number 3 Remote Ping **2** and press Enter.

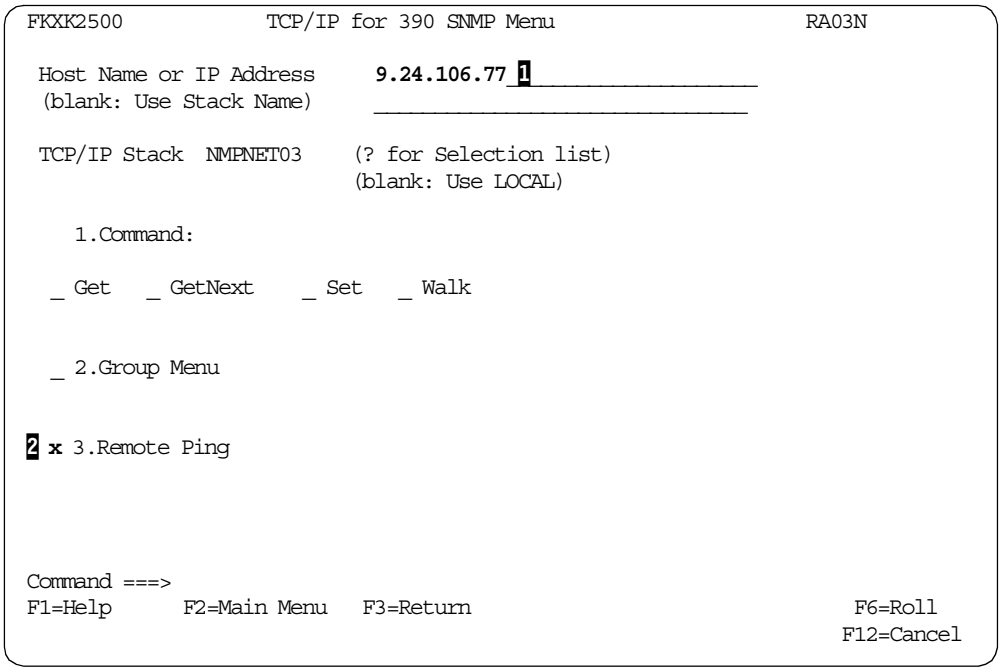


Figure 93. Remote ping

On the following panel, select the TCP/IP stack that you want to use **3** and press Enter again.

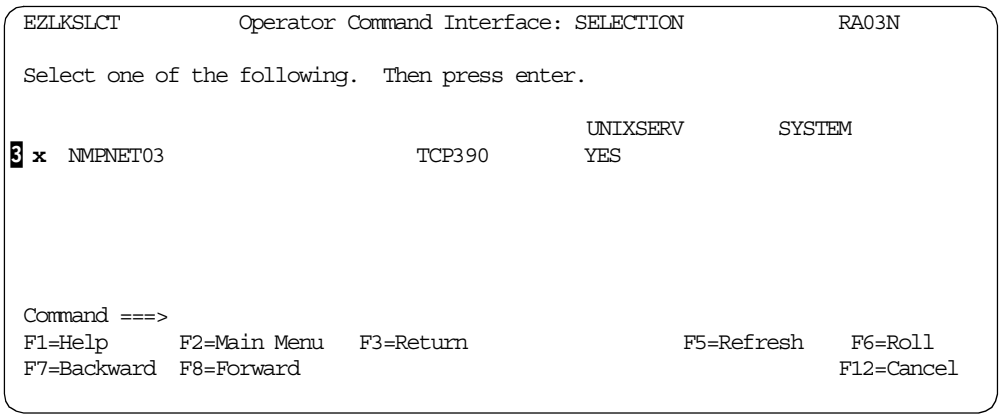


Figure 94. Select the TCP/IP stack

Now, the first panel will be displayed again, with the message

```
FKX910I SERVICE POINT NAME SET. PRESS ENTER TO CONTINUE
```

indicating that the stack has been selected. Then press Enter.

In the following panel, type the name from which you want the ping issued. Remember to use the same notation as in the osnmp.conf file.

```
FKXK2530         TCP/IP for 390 SNMP Remote Ping                 RA03N

 SEND A PING FROM ONE REMOTE HOST TO ANOTHER AND RECEIVE THE RESPONSE
 TIME INFORMATION.

 Send a Ping command
 From:
   v3m39a1_____________________________________________________

 To:
   9.24.106.77_____________________________________________________

 Message Length:  64__

 Timeout Value:   5_


 NOTE: The from Host must be OS/390 v2r5 or higher to support this
       Function


Command ===>
F1=Help      F2=Main Menu   F3=Return                          F6=Roll
                            F9=Options                        F12=Cancel
```

Figure 95. Remote ping panel

Press Enter and the following panel will show the remote ping results:

```
CNMKWIND OUTPUT FROM  Remote Ping to 9.24.106.77          LINE 0 OF 3
*------------------------------ Top of Data --------------------------------*
START OF DATA
ibmMvsRPingResponseTime.64.5.9.24.106.77 = 8
END OF DATA             1  2     3        4
*----------------------------- Bottom of Data ------------------------------*




TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>
```

Figure 96. Remote ping output

This request retrieves the ibmMvsRPingResponseTime MIB object values, which are indexed by three parameters with each part separated by dots(.), as in the following example:

```
ibmMvsRPingResponseTime.packet_size.time_out.ip_address
```

Our sample output in Figure 96 shows:

1 The packet size of the PING request issued was 64 bytes.

2 The timeout value set for this PING request was 5 seconds.

3 9.24.106.77 is the IP address of the remote host that responded to this ping.

4 The round trip response time is 8 millisecs.

4.4 TCP/IP resource management

Extending previous support for managing TN3270 and FTP sessions, any socket connection can now be monitored for availability. Operators can display the session status for any socket connection, including TN3270 sockets, FTP, SMTP, telnet, and Web browsers. If the socket is down, automation can attempt to re-start its associated application, job, or task. IP session management has been extended to include sessions between IP clients and SNA mainframe applications connected through TN3270 servers. Supported servers include IBM's 2210 and 2216 and Cisco's Channel Interface Processors (CIP) and Channel Port Adapters (CPA).

Tivoli NetView for OS/390 can also monitor CS for OS/390 IP resources using either the ping or SNMP operations. Selecting SNMP causes SNMP MIB polling. NetView will then check the interfaces for each resource and compare their administration and operational status. SNMP also enables you to perform MIB thresholding. MIB variables can be checked upon monitoring intervals. NetView checks the MIB variable against the policy definition to determine whether a threshold condition should be raised.

All CS for OS/390 IP resources that should be managed using Tivoli NetView for OS/390 AON panels have to be defined in the AON control file. These resources include the local TCP/IP stack information, the local port numbers on which server applications are running, and IP addresses of the remote IP nodes managed.

4.4.1 Service points management

The session data that is displayed correlates the user's TCP/IP information and SNA information. With this information, you can quickly determine whether a session is hung and take corrective actions. To access the TCP/IP Session Status facility, select “2. MVS IP Session Status” from the MVS TCP/IP Commands Menu AON panel (see Figure 84 on page 115) or issue the `IPSTAT` command on the NCCF command line.

The TCP/IP stack being managed has been defined in the AON control file as shown in Figure 81 on page 113.

Following is the TCP/IP for OS/390 Session Status panel:

```
FKXK2200        TCP/IP for 390 Session Status                        RA03N

 Enter TCP/IP address:  * 1
 or HOSTNAME:  ____________________________________________
               ____________________________________________

   Service Point  System     Host Name             IP Address
2 / NMPNET03      TCPIP      mvs03a                9.24.104.113




Command ===> 3
F1=Help      F2=Main Menu  F3=Return                              F6=Roll
F7=Backward  F8=Forward    F9=Filters   F10=Long Name             F12=Cancel
```

Figure 97. Session status - list of service points

1 Enter an IP address, host name, or wildcard, for example *.

2 Select one, or more, MVS TCP/IP service points by placing any character in front of the field.

3 Press Enter to see the following Session Status Summary panel.

```
FKXK2210          TCP/IP for 390 Session Status                     RA03N

   CLIENT

   *.*.*.*


   Service    Active    IP
   Point      Sessions  Address         HostName
   NMPNET03   7         9.24.104.113    mvs03a
               4



Command ===>
F1=Help      F2=Main Menu  F3=Return    F4=Commands    F5=Refresh    F6=Roll
F7=Backward  F8=Forward    F9=Filters                 F11=Zoom      F12=Cancel
```

Figure 98. Session status - service point

This is the result of the session status search of the selected service point(s). The client field shows the selected search argument with wildcards filled in. Listed are the service point name, IP address of the service point, the hostname of the service point and the number of active sessions found 4. If one or more active sessions are found, then the entry is shown in green. If the service point responded but there were no active sessions found, then the entry is shown in blue. If the service point does not respond, then the number of active sessions is set to N/A and the entry is shown in pink for unknown. The F4 key will take you to the command screen for the service point.

```
FKXK2211         TCP/IP for 390 Session Status                    RA03N
                   SERVICE
   CLIENT - - - -> POINT
                   mvs03a
   *.*.*.*         9.24.104.113
                   TCP/IP STACK      .................................
                                     :                               :
   Service    Active    IP           : NMPNET03    Commands           :
   Point      Sessions  Address      :           _ 1. Pools           :
   NMPNET03   7         9.24.104.113 :             2. Devices         :
                                     :             3. Gateways        :
                                     :             4. Active Sockets  :
                                     :             5. SNMP Cmds       :
                                     :                                :
                                     :                                :
                                     :                                :
                                     :                                :
                                     : F1=Help             F12=Cancel :
                                     :                                :
                                     :................................:
```

Figure 99. Session status - available commands

From this panel the following commands can be issued for the service point selected:

1. Pools: Issue the `NETSTAT POOLSIZE` command and return the response to the NetView screen. This command is not supported by CS for OS/390 V2R5 IP or later releases.
2. Devices: Issue the `NETSTAT DEVICES` command and return the response to NetView.

```
CNMKWIND OUTPUT FROM  COMMAND: NETSTAT DEV ISSUED TO SP NMPN LINE 0 OF 122
*------------------------------ Top of Data -------------------------------*
MVS TCP/IP onetstat CS V2R8       TCPIP Name: TCPIPA           21:18:55
DevName: LOOPBACK          DevType: LOOPBACK  DevNum: 0000
  DevStatus: Ready
  LnkName: LOOPBACK          LnkType: LOOPBACK    LnkStatus: Ready
    NetNum: 0   QueSize: 0   ByteIn: 0000011912   ByteOut: 0000011912
  BSD Routing Parameters:
    MTU Size: 00000             Metric: 00
    DestAddr: 0.0.0.0           SubnetMask: 0.0.0.0
  Multicast Specific:
    Multicast Capability: No

DevName: VIPA3A            DevType: VIPA      DevNum: 0000
  DevStatus: Ready
  LnkName: VIPA3A            LnkType: VIPA        LnkStatus: Ready
    NetNum: 0   QueSize: 0   ByteIn: 0000000000   ByteOut: 0000000000
  BSD Routing Parameters:
    MTU Size: 32768             Metric: 01
    DestAddr: 0.0.0.0           SubnetMask: 255.255.255.0
  Multicast Specific:
    Multicast Capability: No
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>
```

Figure 100. Session status - NETSTAT DEVICES command

3. Gateways: Issue the `NETSTAT GATE` command and return the response to NetView.

```
CNMKWIND OUTPUT FROM  COMMAND: NETSTAT GATE ISSUED TO SP NMP LINE 0 OF 57
*------------------------------ Top of Data -------------------------------*
MVS TCP/IP onetstat CS V2R8       TCPIP Name: TCPIPA           21:19:52
Known gateways:
NetAddress      FirstHop        Link     Pkt Sz Subnet Mask     Subnet Value
----------      --------        ----     ------ -----------     ------------
Default         9.24.104.3      TR1      4082   <none>
9.0.0.0         9.24.104.3      TR1      4082   <none>
9.0.0.0         9.24.104.3      TR1      4082   0.255.254.0     0.1.150.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.3.1.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.3.240.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.12.0.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.12.2.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.240   0.12.3.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.240   0.12.3.16
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.240   0.12.3.32
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.240   0.12.3.48
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.240   0.12.3.64
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.12.6.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.12.13.0
9.0.0.0         9.24.104.3      TR1      4082   0.255.255.0     0.12.14.0
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>
```

Figure 101. Session status - NETSTAT GATE command

4. Active Sockets: Display all active TCP/IP ports (user IDs) on this CS for OS/390 IP stack, and list all user IDs with the status of Listen.

```
FKXK2214         TCP/IP for 390 Session Status
                                                          More   :    +

 Service Point: NMPNET03    Domain: RA03N

 User Id    Conn  Local Socket            Foreign Socket            State

 BPXOINIT   00000 0.0.0.0..10007          0.0.0.0..0                Listen
 FTPD1      00000 0.0.0.0..21             0.0.0.0..0                Listen
 IOASRV     00000 0.0.0.0..8062           0.0.0.0..0                Listen
 NETVA03    00000 0.0.0.0..9998           0.0.0.0..0                Listen
 NETVA03    00000 0.0.0.0..9999           0.0.0.0..0                Listen
 NETVA03    00000 0.0.0.0..4020           0.0.0.0..0                Listen
 TCPIPA     00000 0.0.0.0..7723           0.0.0.0..0                Listen
 TCPIPA     00000 0.0.0.0..8823           0.0.0.0..0                Listen
 TCPIPA     00000 0.0.0.0..6623           0.0.0.0..0                Listen
 TCPIPA     00000 0.0.0.0..9923           0.0.0.0..0                Listen
 TCPIPA     00000 0.0.0.0..1025           0.0.0.0..0                Listen



Command ===>
F1=Help      F2=Main Menu   F3=Return    F4=Commands                F6=Roll
F7=Backward  F8=Forward                                            F12=Cancel
```

Figure 102. Session status - active sockets display

5. SNMP Cmds: Go to SNMP Functions for this CS for OS/390 IP stack. You will see an AON panel similar to Figure 85 on page 116.

4.4.2 Session status management

In the Session Status AON panel, you will see all the active connections destined to the applications defined with the IPPORT statement in the AON control file. We have defined a TN3270 server and FTP server running on the local TCP/IP stack as shown below:

```
IPPORT TN3270,SP=NMPNET03,                    1
               PORT=23,                       2
               PROTOCOL=TCP,                  2
               TCPNAME=TCPIPA,                3
               INTVL=00:10,                   4
               FORMAT=PORT,
               STATUS=NORMAL,
               DESC="Telnet Socket Pointer"
*
IPPORT FTPSRV2,SP=NMPNET03,                   1
            PORT=21,                          2
            PROTOCOL=TCP,                     2
            TCPNAME=FTPDA1,                   3
            INTVL=00:10,                      4
            FORMAT=PORT,
            STATUS=NORMAL,
            DESC="FTP Server on RA03"
```

Figure 103. Sample IPPORT definitions in the AON control file

The following keywords have been defined for our installation:

- The unique name of the resources and the NetView service point name 1 on which servers should run. The service point name has be synonymous with the MVS service point definition.
- The port number and transport protocol name, TCP or UDP, that the server application uses 2 .
- 3 The job name of server applications monitored. For the OS/390 TN3270 server which is running inside the TCP/IP address space, we used the same name as TCP/IP itself. For FTP server monitoring, the job name of the listener task, which is FTPDA1 in our configuration, has been specified.

  Note: Since the MVS started procedure for the FTP server in our environment is named FTPDA, TCP port 21 is listened on by the FTPDA1 address space (FTP listener), which is a child process of FTPDA.
- 4 The monitoring interval in hh:mm format.

For a complete description of each keyword, refer to *Tivoli NetView for OS/390 Administration Reference Version 1 Release 3*, SC31-8222.

The following Session Status panel displays three TN3270 sessions from workstations 9.24.106.77 and 172.16.250.3 to TSO being managed by the MVS TCP/IP service point named NMPNET03. To display this panel, press PF11 or Enter after moving the cursor on the resource you want to choose from panel FKXK2210 (Figure 98 on page 125).

```
FKXK2220          TCP/IP for 390 Session Status                  RA03N
                  SERVICE
   CLIENT - - - -> POINT
                  NMPNET03                    Active Sessions  6     1
   *.*.*.*        9.24.104.113              Filtered Sessions  6     2
                  TCP/IP STACK
Client                                                          Logical  Appl.
Port   Client Address   Type      Status    Send     Rec       Unit SLU (PLU)
1030   9.24.104.149     FTPSRV2   ESTABLSH  00000140 00000000
3188   9.24.106.31      TN3270    ESTABLSH  00001120 00015993 RA03TN04 RA03T03
3187   9.24.106.31      TN3270    ESTABLSH  00000099 00000530 RA03TN02 RA03T04
1474   9.24.106.113     FTPSRV2   FINWAIT2  00000140 00000000
1473   9.24.106.113     FTPSRV2   ESTABLSH  00000140 00000000
1411   9.24.106.113     TN3270    ESTABLSH  00000062 00000341 RA03TN01
1412   9.24.106.113     TN3270    ESTABLSH  00000099 00000530 RA03TN03 RA03T05
 3                                                                4       5


Command ===>
F1=Help      F2=Main Menu   F3=Return   F4=Commands   F5=Refresh     F6=Roll
F7=Backward  F8=Forward     F9=Filters                              F12=Cancel
```

Figure 104. Session status - list of session

This screen shows the active sessions established between the selected service point and clients whose IP addresses match the parameters set by the session status filters. See 4.4.3, “Sample session filtering: Telnet and FTP” on page 136 for further infomation on session filtering.

The Active Sessions field 1 contains the total number of active sessions for this service point. The Filtered Sessions field 2 contains the number of sessions matching the filter criteria.

Listed are the client port, the IP address or hostname for the session, the session type, status, send/receive counts, and for telnet sessions, the LU name and MVS application name.

You can issue commands for the session 3, logical unit 4, or application 5 by moving your cursor to the field and pressing PF4 (Commands). The following sections describe the details about these commands.

4.4.2.1 Issue commands for the selected connection

By pressing PF4 when a cursor is located on the Client Port field, you will see panel FKXK2221, in which several TCP/IP commands can be issued for the selected connection.

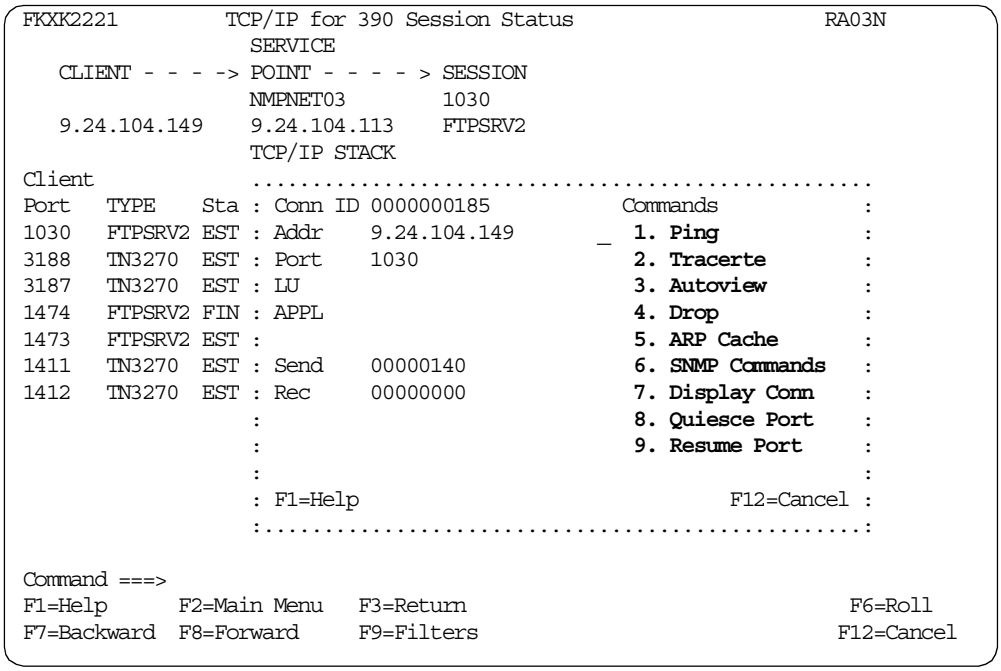


Figure 105. Session status - TCP/IP commands for the selected connection

The available commands are:

1. Ping: Issue the PING command to this TCP/IP resource and return the response to the NetView screen.

```
CNMKWIND OUTPUT FROM  COMMAND: PING 9.24.106.31 ISSUED TO SP LINE 0 OF 2
*------------------------------ Top of Data -------------------------------*
CS V2R8: Pinging host 9.24.106.31
Ping #1 response took 0.013 seconds.
*---------------------------- Bottom of Data ------------------------------*
```

Figure 106. Session status - ping command

2. Tracerte: Issue TRACERTE to the IP address of this TCP/IP resource and return the response to the NetView screen.

```
CNMKWIND OUTPUT FROM  COMMAND: TRACERTE 9.24.106.31 ISSUED T LINE 0 OF 4
*------------------------------ Top of Data -------------------------------*
Traceroute to 9.24.106.31 (9.24.106.31).
Use escape C sequence to interrupt
1 POK2210A.itso.ral.ibm.com (9.24.104.3)  12 ms  11 ms  11 ms
2 WTR05208.itso.ral.ibm.com (9.24.106.31)  8 ms  11 ms  11 ms
*---------------------------- Bottom of Data ------------------------------*
```

Figure 107. Session status - tracerte command

3. Autoview: Go to AON Autoview for this TCP/IP resource.

```
FKXKVIEW        TCP/IP Automation: AutoView                        RA03N


Resname . . . . . 9.24.106.31       Restype . . . . . IPHOST
Service Point . . . . . NMPNET03    IP Address  . . . . . .
Hostname  . . . . . 9.24.106.31
Links . . . . . .

Explicitly Defined. . . NO          Automation Flag . . . . ON
Option  . . . . . IP390             In Recovery . . . . . NO
Status  . . . . . UNKNOWN           In Active Monitoring. . NO
Automation Status . . . UNKNOWN     Name Server Status. . . UNKNOWN

Pingstats . . . . . NONE

DDF message . . . . . NONE




Command ===>
F1=Help      F2=Main Menu   F3=Return    F4=Commands   F5=Refresh    F6=Roll
                                                                   F12=Cancel
```

Figure 108. Session status - AutoView

4. Drop: Issue the `NETSTAT DROP` command for this session ID which will end this connection. After the command is completed, you will see a message similar to the following:

   ```
   FKX611I SESSION 0000000057 WAS SUCCESSFULLY DROPPED
   ```

5. ARP Cache: Issue the NETSTAT ARP command for this TCP/IP resource and return the response to the NetView screen.

```
CNMKWIND OUTPUT FROM  COMMAND: NETSTAT ARP 9.24.104.149 ISSU LINE 0 OF 5
*------------------------------ Top of Data --------------------------------*
MVS TCP/IP onetstat CS V2R8       TCPIP Name: TCPIPA          15:20:29
Querying ARP cache for address 9.24.104.149
Link: TR1              IBMTR: 400052005042
Route info: 0220

*-----------------------------BottomofData-------------------------------*
```

Figure 109. Session status - NETSTAT ARP command

6. SNMP Commands: Go to the SNMP Commands Menu AON panel for this TCP/IP resource. See Figure 85 on page 116.
7. Display Conn: Issue the `Display TCPIP,,Telnet,CONN` command to display detailed TN3270 connection information. This option is valid only for TN3270 connections.

```
CNMKWIND OUTPUT FROM  MVS DISPLAY TCPIP,TCPIPA,T,CONN,CONN=0 LINE 0 OF 15
*------------------------------ Top of Data -------------------------------*
EZZ6065I TELNET CONNECTION DISPLAY 583
  CONN: 00000185    IPADDR..PORT: 9.24.106.113..1411
  HOSTNAME: NO HOSTNAME
  CONNECTED: 14:08:35  04/03/2000  STATUS: SESSION PENDING
  PORT:    23  ACTIVE   BASIC           ACCESS: NON-SECURE
  PROFILE ID: CURR   PROFILE OPTIONS: -------W-----
  PROTOCOL: TN3270E  LOGMODE:          DEVICETYPE: IBM-3278-2-E
    OPTIONS: ETET---  3270E FUNCTIONS: BSR--
  USSTABLE: **N/A**     HN/IPGROUP: **N/A**
  LUNAME: RA03TN01      TYPE: TERMINAL
    LU/PRTGROUP: LU1         HN/IPGROUP: IP1
  APPLID:
    DEFAULTAPPL HN/IPGROUP: **N/A**
    RESTRICTAPPL USERID: **N/A**
13 OF 13 RECORDS DISPLAYED
*----------------------------- Bottom of Data -----------------------------*
```

Figure 110. Session status - display telnet connection information

8. Quiesce Port: Issue the `Vary TCPIP,,Telnet,QUIESCE` command against the TN3270 port to prevent new connections from being established. This option is valid only for the TCP port on which a TN3270 server is running.
9. Resume Port: Issue the `Vary TCPIP,,Telnet,RESUME` command against the port to allow new connections to be established again. This option is valid only for the TCP port on which a TN3270 server is running.

4.4.2.2 Issue command to LUs used for a TN3270 connection

If the cursor is placed on the LU (PLU) or APPL (SLU) resource, pressing PF4 gives you the panel FKXK2225 or FKXK222A respectively. From these panels, you can manage secondary LUs or application LUs (primary LUs) by issuing commands.

```
FKXK2225         TCP/IP for 390 Session Status                    RA03N
                   SERVICE
   CLIENT - - - -> POINT - - - - > SESSION - - -> LU
                   NMPNET03        2894
   9.24.106.77     9.24.104.113    TN3270         RA03TN01
                   TCP/IP STACK
Client                    ..........................................
Port   TYPE    Status     : Lu    RA03TN01       Commands             :
2902   FTPSRV2 ESTABLSH   : Appl  RA03T04       _ 1. Display LU      :
2879   TN3270  ESTABLSH   :                       2. D NET IPADDR     :
2901   FTPSRV2 FINWAIT2   :                       3. SNA Helpdesk     :
2894   TN3270  ESTABLSH   :                       4. Autoview         :
                          :                       5. NLDM Sess List   :
                          :                                           :
                          :                                           :
                          :                                           :
                          : F1=Help                        F12=Cancel :
                          :                                           :
                          :...........................................:

Command ===>
F1=Help      F2=Main Menu   F3=Return                          F6=Roll
F7=Backward  F8=Forward     F9=Filters                       F12=Cancel
```

Figure 111. Session status FKXK2225 panel - LU commands

The commands available from the FKXK2225 panel are:

1. Display LU: Issue the VTAM `DISPLAY` command to display the status of the secondary LU selected. VTAM message IST453I will be received for resources that are not known to this VTAM.

```
CNMKWIND OUTPUT FROM  SNA DISPLAY OF RA03TN01                LINE 0 OF 24
*------------------------------ Top of Data --------------------------------*
IST097I  DISPLAY  ACCEPTED
IST075I  NAME = USIBMRA.RA03TN01 , TYPE = DYNAMIC APPL
IST486I  STATUS= ACT/S     , DESIRED STATE= ACTIV
IST1447I  REGISTRATION TYPE = CDSERVR
IST1629I  MODSRCH = NEVER
IST977I  MDLTAB=***NA*** ASLTAB=***NA***
IST861I  MODETAB=ISTINCLM USSTAB=***NA*** LOGTAB=***NA***
IST934I  DLOGMOD=***NA*** USS LANGTAB=***NA***
IST1632I  VPACING =  7
IST597I  CAPABILITY-PLU ENABLED  ,SLU ENABLED  ,SESSION LIMIT 00000001
IST231I  APPL     MAJOR NODE = TELAPPL
IST1425I  DEFINED USING MODEL RA03TN??
IST654I  I/O TRACE = OFF, BUFFER TRACE = OFF
IST1500I  STATE TRACE = OFF
IST271I  JOBNAME = TCPIPA  , STEPNAME = TCPIPA  , DSPNAME = IST7B41D
IST228I  ENCRYPTION = OPTIONAL    , TYPE = DES
IST1563I  CKEYNAME = RA03TN01 CKEY = PRIMARY   CERTIFY = NO
IST1552I  MAC = NONE        MACTYPE = NONE
IST1050I  MAXIMUM COMPRESSION LEVEL - INPUT = 0   , OUTPUT = 0
IST1633I  ASRCVLM =  1000000
TO SEE YOUR KEY SETTINGS, ENTER 'DISPFK'
CMD==>
```

Figure 112. Session status - resource display

2. D NET IPADDR: Issue the VTAM `D NET,IPADDR` command to display TN3270 session information. This function is not available for a TN3270 server running on an IBM2210/2216 router or Cisco router.
3. SNA Helpdesk: Go to the SNA Help Desk for this LU.

   Note: If you use SNAHD to recycle the LU, this will disconnect the session.

```
FKVK1000                  SNA Automation: Help Desk

 ****  *   **   **      **  **         ***         *****               **
**   * **  **  ****     **  **          **         **  **              **
 **    *** ** **  **    **  **  ****    **   *****  **  **  ****   ***** ** **
  **   ****** ******    ****** **  **   **   **  ** **  ** **  ** **     ***
   **  ** *** **  **    **  ** ******   **   **  ** **  ** ******   **   ****
*   ** **  ** **  **    **  ** **       **   **  ** **  ** **         ** ** **
 ****  **   * **  **    **  **  ****   ****  *****  *****   ****  *****  **  **
                                              **
                                              **

            Enter name...:  RA03TN01

            Select option:  _  1. Recycle resource
                               2. Problem Determination
                               3. NetView Access Services User ID
                               n. NetView HelpDesk



EZL910I ENTER A SELECTION
Command ===>
```

Figure 113. Session status - SNA help desk

4. Autoview: Go to AON Autoview for this SNA resource

```
EZLKVIEW              AON: AutoView                                RA03N


Resname . . . . . . . . RA03TN01
Restype . . . . . . . . LU
Option  . . . . . . . . AON
Status  . . . . . . . .
Automation Status . . . UNKNOWN
Automation Flag . . . .
DDF message . . . . . . NONE



Resource Definitions _ 1. Automation          (RECOVERY DEFAULTS)
                       2. Thresholds          (THRESHOLDS DEFAULTS)
                       3. Active Monitoring   (ACTMON DEFAULTS)
                       4. Monitor Intervals   (MONIT DEFAULTS)
                       5. Timer               (NONE)
                       6. Display Network LOG Information For RA03TN01


Command ===>
F1=Help      F2=Main Menu   F3=Return               F5=Refresh      F6=Roll
                                                                   F12=Cancel
```

Figure 114. Session status - AON autoview

5. NLDM Sess List: Go to the NLDM session list for this LU and Appl.

```
NLDM.SESS                                                          PAGE    1
                                SESSION LIST
NAME: RA03TN01 RA03T04                                          DOMAIN: RA03N
--------------------------------------------------------------------------------
      ***** PRIMARY *****  **** SECONDARY ****
 SEL#   NAME   TYPE  DOM     NAME   TYPE  DOM     START TIME       END TIME
 ( 1) RA03T04  LU   RA03N  RA03TN01 LU   RA03N  03/06 15:01:58  *** ACTIVE ***









END OF DATA
ENTER SEL# (CONFIG), SEL# AND CT (CONN. TEST), SEL# AND STR (TERM REASON)
CMD==> 1
```

Figure 115. Session status - session list

By selecting the session number on this panel, more detailed session information is displayed as shown in Figure 116.

```
NLDM.CON                SESSION CONFIGURATION DATA                 PAGE    1
------------- PRIMARY ---------------+-------------- SECONDARY --------------
NAME RA03T04   SA 0003  EL 00000150   |   NAME RA03TN01  SA 0003  EL 0000014F
---------------------------------------+----------------------------------------
DOMAIN RA03N       PCID USIBMRA.RA03M.C7335B7CAB58B179          DOMAIN RA03N
              +-------------+                 +-------------+
RA03M         |   CP/SSCP   | ---         --- |   CP/SSCP   | RA03M
ISTPUS03(0000)| SUBAREA PU  |                 | SUBAREA PU  | ISTPUS03(0000)
              +------+------+                 +------+------+
                     |                               |
              +------+------+                 +------+------+
RA03T04 (0150)|     LU      |                 |     LU      | RA03TN01(014F)
              +-------------+                 +-------------+
                            APPNCOS N/A
                            SUBACOS N/A
                            LOGMODE SNX32702
                            PADJ CP N/A
                            SADJ CP N/A




SELECT PT, ST (PRI, SEC TRACE), RT (RESP TIME), P, AR, FC
CMD==>
```

Figure 116. Session configuration data panel

From the FKXK222A panel, that is shown by pressing PF4 when a cursor is placed at the application LU field, the same command can be issued against a selected application LU.

4.4.3 Sample session filtering: Telnet and FTP

NetView provides the ability to display TCP/IP session status information for selected stacks. The session status filters enable you to display specific resources. These filters are set by task. You can invoke the Session Status Filters panel using the PF9 key on selected TCP/IP for OS/390 panels. In the following example we will invoke the Session Status Filter panel and we'll choose to show only the FTP connections directed to application RA03TN01.

Before activating session filtering, we have to look at five active connections listed in the Session Status panel (see Figure 117), both FTP and telnet. Press the PF9 key to display the Session Status Filters panel.

```
FKXK2220           TCP/IP for 390 Session Status                  RA03N
                  SERVICE
  CLIENT - - - -> POINT
                  NMPNET03                    Active Sessions  5
  9.24.106.77     9.24.104.113              Filtered Sessions  5
                  TCP/IP STACK
Client                                                      Logical  Appl.
Port   Client Address   Type      Status    Send     Rec     Unit SLU (PLU)
1597   9.24.106.77      FTPSRV2   ESTABLSH  00000139 00000000
1598   9.24.106.77      FTPSRV2   ESTABLSH  00000139 00000000
1599   9.24.106.77      FTPSRV2   ESTABLSH  00000139 00000000
1604   9.24.106.77      TN3270    ESTABLSH  00000032 00000311 RA03TN01
1601   9.24.106.77      TN3270*   TIMEWAIT  00000073 00000319 RA03TN01




Command ===>
F1=Help      F2=Main Menu   F3=Return   F4=Commands   F5=Refresh     F6=Roll
F7=Backward  F8=Forward     F9=Filters                               F12=Cancel
```

Figure 117. Session status main panel

The resulting panel shows the session status filter settings currently defined for the operator and allows the settings to be changed as necessary. To change settings, tab to any entry that is to be changed and type the new information. When Enter is pressed, the changes will be processed.

```
FKXKSSF0        TCP/IP for 390 Session Status Filters            RA03N

The current filter and sort settings are shown.
1
Type the data to be displayed in one or more fields.
Client IP                                                    Logical
Port   Address          Type      Status   Send      Rec      Unit     Appl.
n/a    *                FTPSRV2   *        n/a       n/a      *        *
Type an action code to define the logical operator for the search:
2  1 1 OR
     2 AND
3
Type numbers from  1    to  5    to define the column order for sorting.
Client IP                                                    Logical
Port   Address          Type      Status   Send      Rec      Unit     Appl.
n/a    1                5         4        n/a       n/a      3        2
Type an action code to define the sort order.
4 1 1 Ascending order
     2 Descending order

Command ===>
F1=Help                    F3=Return                          F6=Roll
                           F9=Defaults                        F12=Cancel
```

Figure 118. Session Status Filter panel

1 The first section shows the data to be displayed on the Session Status panel. A blank or "*" in an input field indicates that no filtering of data is to be performed on that field. We only entered FTPSRV2 in the Type field but we could have made multiple entries.

2 The second section shows the logical operator to be used for the filtering criteria set in the first section. The "OR" operator causes sessions that match any of the specified criteria to be displayed. The "AND" operator causes sessions that match all of the specified criteria to be displayed. We left the default value.

3 The third section shows the sort order to be used when the session status is displayed. These fields accept only the numbers from one to five and blank as input. At least one field must be specified. The input must be sequential starting with 1. If an operator enters 1 in the Appl. field and 2 in the IP Address field, the Session Status display will be sorted by the Appl. field first. If there are identical Appl. entries, they will then be sorted by the IP Address field. We left the default value.

4 The last section shows the order to be used for the sort.

Return to the session status main menu by pressing PF3.

```
FKXK2220           TCP/IP for 390 Session Status                  RA03N
                   SERVICE
   CLIENT - - - -> POINT
                   NMPNET03                   Active Sessions  4
   *.*.*.*         9.24.104.113             Filtered Sessions  3
                   TCP/IP STACK
Client                                                        Logical  Appl.
Port   Client Address   Type      Status    Send     Rec      Unit SLU (PLU)
1597   9.24.106.77      FTPSRV2   FINWAIT2  00000139 00000000
1598   9.24.106.77      FTPSRV2   FINWAIT2  00000139 00000000
1599   9.24.106.77      FTPSRV2   FINWAIT2  00000139 00000000



Command ===>
F1=Help      F2=Main Menu   F3=Return   F4=Commands   F5=Refresh     F6=Roll
F7=Backward  F8=Forward     F9=Filters                              F12=Cancel
```

Figure 119. Session status panel after filtering

As requested, the resulting panel lists only the FTP connections currently active for our client IP address.

4.4.4 Proactive monitoring and MIB thresholding

When defining resources to be monitored you can ping a resource or use SNMP functions. Selecting ping causes NetView to ping the resource to determine whether it is active. Selecting SNMP causes SNMP MIB polling. NetView will then check the interfaces for each resource and compare their administration and operational status. SNMP also enables you to perform MIB thresholding. MIB variables can be checked upon monitoring intervals. NetView checks the MIB variable against the policy definition to determine whether a threshold condition should be raised. For example, you can monitor a router and define key performance MIB variables that are checked at each monitoring interval. User exits are provided for both MIB polling and MIB thresholding. This enables more extensive MIB queries for further analysis or automation.

Using AON proactive monitoring, you can use SNMP requests instead of ping commands. SNMP proactive monitoring provides MIB polling functions. MIB polling queries the interface table of the device being monitored. If one or more interfaces have an incorrect status, AON/TCP will send a notification. You can code a user exit for further processing.

MIB thresholding can occur while a device is being proactively monitored. MIB thresholding will query MIB variables that you define and compare their expected values with the actual values. AON/TCP will compare less than, less than or equal, equal, greater than or equal, and greater than conditions. When a threshold is exceeded, AON/TCP will send a notification. You can code a user exit for further processing.

As an example, you can use the MIB polling and thresholding functions for a router. MIB polling will detect failed interfaces on the router. With MIB thresholding, you can define performance-related MIB variables and their thresholds.

In our implementation, we defined the following monitoring rules and MIB thresholds for the IPHOST MVS03A in DSIPARMU(FKXCFG01):

```
ACTMON STACKS,OPTION=IP390,INTVL=00:30,STATUS=NORMAL,
       FORMAT=SNMP                                  1
*
IPHOST MVS03A,OPTION=IP390,SP=NMPNET03,
       ACTMON=STACKS,
       MIBVAR1=(tcpActiveOpens.0,LT,5000),          2
       MIBVAR2=(tcpInErrs.0,GT,5),                  2
       MIBVAR3=(ipInDiscards.0,EQ,1),               2
       MIBVAR4=(ipRoutingDiscards.0,GE,3),          2
       STATUS=(NORMAL,TRESH*,DEGR*),
       HOSTNAME=MVS03A                              3
```

Figure 120. AON control file definitions for MIB thresholding

As you can see, the keyword `FORMAT=SNMP` 1, tells AON to monitor resource MVS03A with SNMP commands. If SNMP is used, a utility FKXECMAN is used to GET SNMP variables for that resource and the community name used is read from the /etc/netview/fkxcm.data file in the UNIX environment.

FKXECMAN is an OS/390 UNIX executable module that is used to retrieve MIB variables using the community name selected. It does the following:

1. Reads the file /etc/netview/fkxcm.data
2. Matches the resource name to the supplied community name
3. Appends the community name to the SNMP request
4. Issues the SNMP request

Refer to *Tivoli NetView for OS/390 Administration Reference Version 1 Release 3*, SC31-8222 for complete descriptions of the IPHOST statement, and *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236 for further information on the FKXECMAN utility.

This is the /etc/netview/fkxcm.data file used in our project:

```
************************************************************************
* Licensed Materials - Property of Tivoli Systems                      *
* 5697-B82 (c) Copyright Tivoli Systems 1999                           *
* All rights reserved.                                                 *
*                                                                      *
* US Government Users Restricted Rights - Use, duplication or          *
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.    *
************************************************************************
*COMMUNITY******  *IPADDRESS***** *HOSTNAME
*public           *9.24.104.113   *MVS03A     5
publicv2c         9.24.104.113    MVS03A      4
publicv1          9.24.104.113    MVS03A1
public            9.24.104.26     AIXKEN
```

Figure 121. Sample /etc/netview/fkxcm.data file

The host name configuration in both the IPHOST definition in the AON control file 3 and the /etc/netview/fkxcm.data file 4 are case-sensitive. We recommend that you use uppercase host names for both configurations.

If you want to comment out lines in fkxcm.data, all host names in those lines have to be prefixed by asterisks (*). Otherwise, the host name matches the value that is being used for the community name search, and then an invalid community name (possibly prefixed by an asterisk that is added for commenting out the line)

will be picked up. See the line starting with `*public` in Figure 121 5, to see how we used asterisks to comment out one line properly.

The keywords `MIBVARx` 2, used in this example, are used to set thresholds for four different MIB variables. If one or more values of these variables meet the conditions specified, the resource status will change to THRESH.

The available relational operators are:

Table 23. Relational operations

Operator	Description
EQ	Equal
LT	Less Than
LE	Less Than or Equal
GT	Greater Than
GE	Greater Than or Equal
NE	Not Equal

By selecting 7. IP Resource Manager at the MVS TCP/IP Commands Menu (panel FKXK2000, see Figure 84 on page 115), you can reach the Resource Management AON panel, where you can manage and/or monitor IP resources defined in the AON control file. The supported operations in this panel are:

- Add IP resources to be monitored dynamically.
- Display and change the configuration for IP resources. The threshold values also can be changed dynamically.
- Delete IP resources that no longer need to be monitored.
- Start and stop monitoring for the selected IP resource.

From the OS/390 UNIX shell environment, you can issue exactly the same FKXECMAN command that is executed behind the AON/TCP panels to monitor selected MIB variables. We issued the following command to retrieve the MIB variables that are being monitored with MIB thresholding.

```
LUCY @ RA03:/u/lucy>fkxecman -v -h MVS03A GET  TcpActiveOpens.0  tcpInErrs.0
ipInDiscards.0 ipRoutingDiscards.0
tcpActiveOpens.0 = 2310 3
tcpInErrs.0 = 0
ipInDiscards.0 = 0
ipRoutingDiscards.0 = 0
```

In our example, the value of `tcpActiveOpens.0` 3 is 2310, and meets the specified condition, which is less than 5000.

The Resource Management panel then shows the status of THRESH for this resource as shown in Figure 122 on page 141.

```
FKXK2700          TCP/IP for 390 Resource Management

 Select an AON control file or monitoring command and press ENTER
 1 = ADD/START 2 = DISPLAY/CHANGE  3 = DELETE  4 = START  5 = STOP
                     Resource    TCP/IP      Actmon
     Resource        Type        Stack       Definition    M Status
     --------------- ----------- ----------- ------------- - -------
 _   DSITCPIP        IPPORT      NMPNET03                  R DOWN
 _   DSIWBTSK        IPPORT      NMPNET03                  R DOWN
 _   FTPSRV2         IPPORT      NMPNET03                  A NORMAL
 _   KENAIX          IPHOST      NMPNET03    ALLHOSTS      A NORMAL
 _   MVS03A          IPHOST      NMPNET03    STACKS        N THRESH
 _   NMPNET03        TCP390      NMPNET03                  A NORMAL
 _   TN3270          IPPORT      NMPNET03                  A NORMAL



Command ===>
F1=Help      F2=Main Menu   F3=Return   F4=Commands    F5=Refresh    F6=Roll
F7=Backward  F8=Forward     F9=Display Options                     F12=Cancel
```

Figure 122. Resource management panel

Chapter 5. NetView Management Console (NMC)

The NetView Management Console (NMC) is a function of the Tivoli NetView for OS/390 program that provides a graphical topological presentation of the network controlled by NetView. It enables you to manage that network interactively.

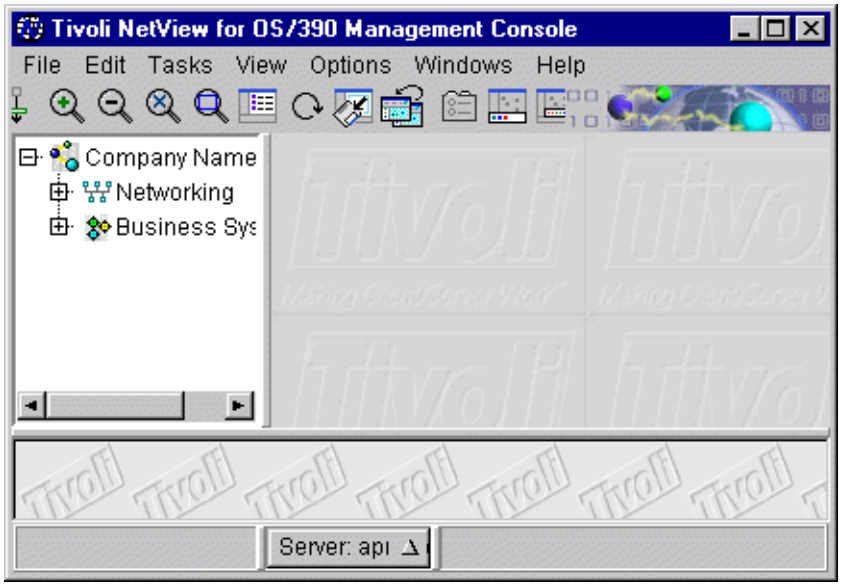


Figure 123. NetView Management Console

5.1 NMC Overview

NMC consists of a server and a Java-based console, which are generically referred to in this book and the online help as the topology server and topology console. The topology console graphically displays systems and networking information provided by Tivoli NetView for OS/390. This information is displayed as Resource Object Data Manager (RODM) based views and is only available if there is a conversation set up between the topology server and NetView.

NMC enables you to:

- Monitor and control large portions of complex business systems.
- View the topology and connectivity of your network graphically.
- Monitor the overall state of a network or a portion of a network through aggregates, which represent the combined status of a group of related applications and resources.
- Navigate easily from an aggregate to a real resource that is failing.
- Mark resources for your own purposes, for example, to show that they are being serviced.
- Display a list of events received for a selected resource.
- Issue predefined commands from context menus, or use a command line to issue your own commands.
- Stop and restart selected resources.

- Specify which resources are critical to your network so that operators are notified when they are inactive.

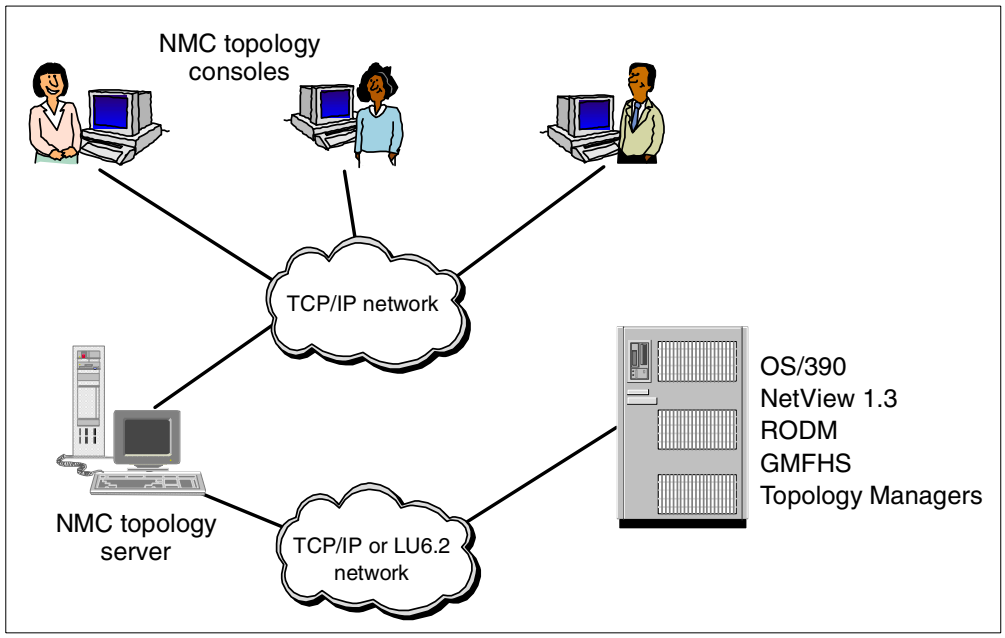


Figure 124. NMC overview

The topology server is connected to Tivoli NetView for OS/390 with the NETCONV command and can use a TCP/IP or LU 6.2 session. Once connected the server receives all available network views from NetView and makes them available to all the topology consoles (NMCs). The topology server and the topology console can also be installed on the same workstation.

5.1.1 Required environment

In this section we will list the environment required by NMC.

5.1.1.1 Supported platforms

Topology server requires one of the following:

- AIX 4.2 (5765-655) or later
- Windows 2000
- Windows NT 4.0 with service pack 3 or later
- OS/2 WARP 4.0 with fixpack 9 or later

You may install topology console on the following:

- AIX 4.2 (5765-655) or later
- OS/2 WARP 4.0 with fixpack 5 or higher
- Windows NT 4.0 or later
- Sun Solaris 2.6 or later
- HP-UX 10.2 or higher
- Windows 95 or 98

Please refer to *Tivoli NetView for OS/390 Planning Guide Version 1 Release 3*, GC31-8226 for information about hardware and software prerequisites.

5.1.1.2 Communication requirements

The Topology server requires LU 6.2 or TCP/IP connectivity to communicate with Tivoli NetView for OS/390. If TCP/IP is used, TCP/IP V3R2 or higher is required for OS/390.

The NMC console requires TCP/IP connectivity to communicate with the topology server.

5.1.1.3 Java

Java for OS/390 at the Java Development Kit (JDK) 1.1.6 or 1.1.8 level is required by Java SNMP services that includes the MIB browser and Java SNMP manager.

Java code has been included in the base NMC code so there is no need to install it on the workstation. No additional customization steps are needed for Java setup.

For more information, see the readme files for NMC server and client.

5.2 NMC installation on your workstation

The topology server interacts with RODM and provides information for display on the NMC client. The topology server furnishes the NMC client with a set of tasks that are applicable against a resource. These tasks appear in the context-sensitive menus on the console. The topology server also stores files on behalf of the NMC client, including icons, backgrounds, help files, log files, customization views, and settings.

We installed the code from Internet linking at this URL:

```
http://www.tivoli.com/nv390_supported
```

On the first page, namely “Tivoli NetView for OS/390 - Supported Functions”, select the **NetView Management Console (NMC)** by clicking it with the mouse.

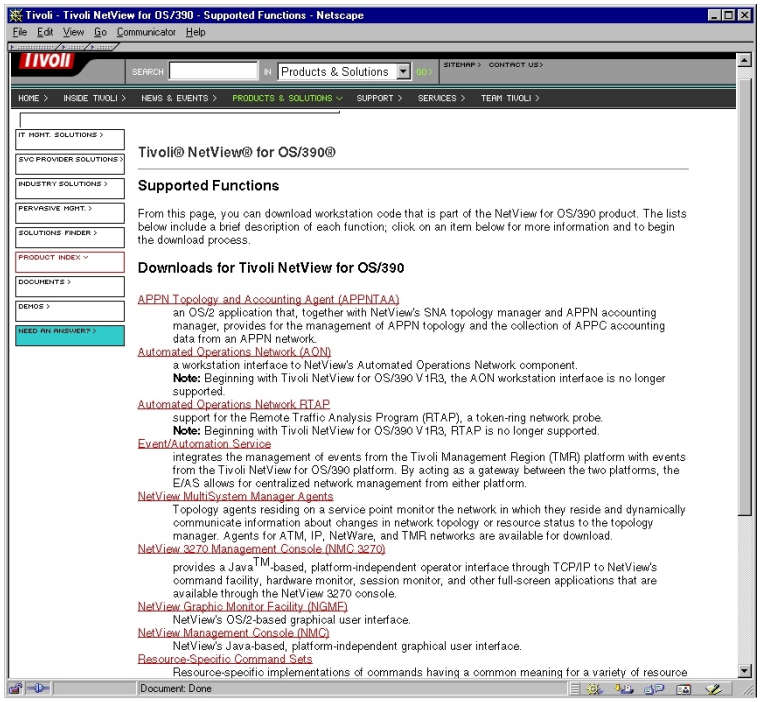


Figure 125. Tivoli NetView for OS/390 Supported Functions main page

At the "NetView Management Console (NMC)" page, select **NetView Management Console Topology Server**.

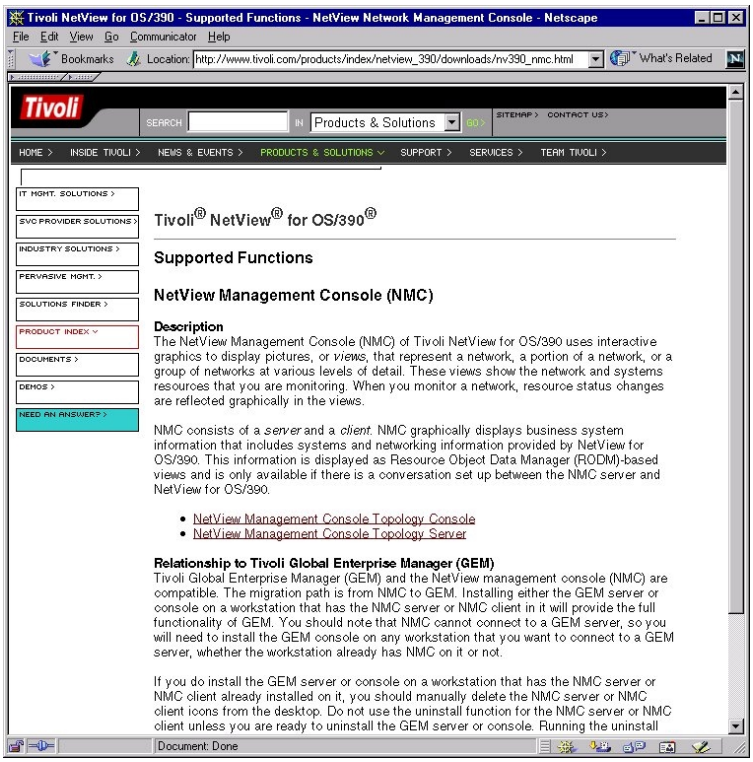


Figure 126. Tivoli NetView for OS/390 Supported Functions - NetView Management Console

Select **NetView for OS/390 V1R3**.

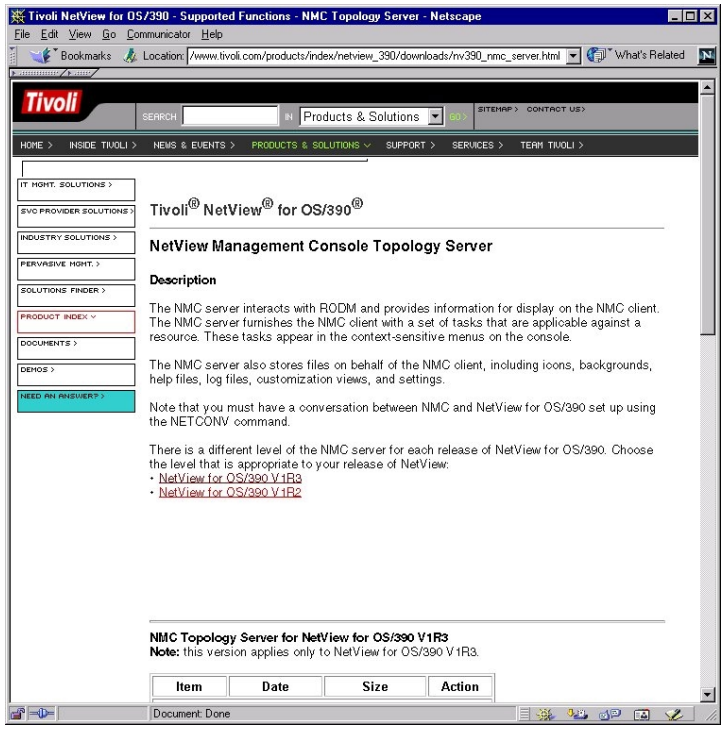


Figure 127. Tivoli NetView for OS/390 Supported Functions - NMC Topology Server

Choose your workstation environment. and click **Download**.

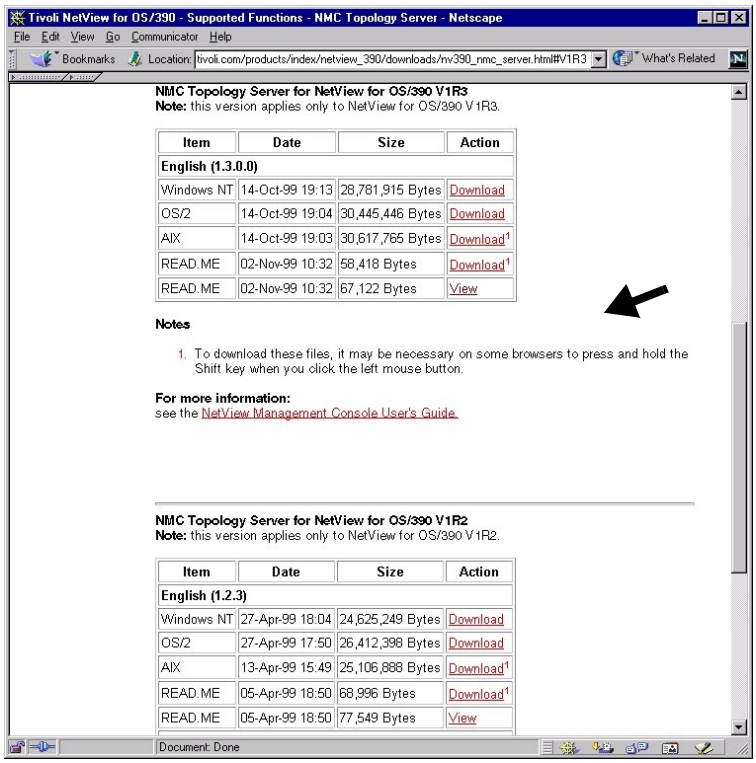


Figure 128. Tivoli NetView for OS/390 Supported Functions - NMC Topology Server download

Save the file on a temporary directory on your hard disk and execute it. Follow the installation instructions and leave all the default values regarding directory names. The NMC topology server will be installed on your workstation.

For the NMC topology console installation, follow the same instructions as those for the NMC topology server starting from the same download page (see Figure 126 on page 146). Download the code, and then install it on your workstation.

The Resource Specific Command Set is also required for NMC to work correctly. Go back to the “Tivoli NetView for OS/390 Supported Functions” main page, and select **Resource-Specific Command Sets** to move to its installation page (see Figure 129 on page 148).

Note: Remember to download the READ.ME file for each product and store it on your hard disk. These files contains very helpful hints and tips about NMC products.

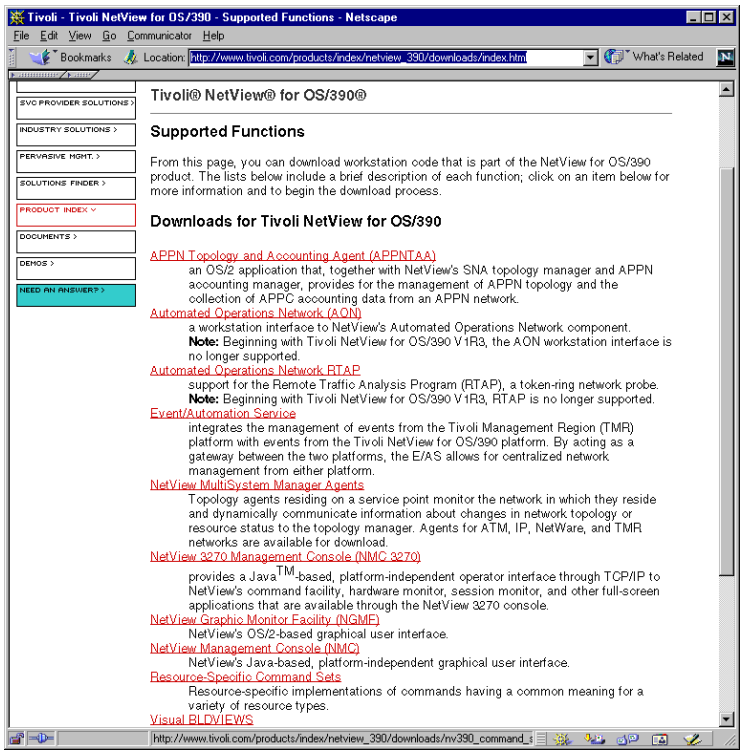


Figure 129. Tivoli NetView for OS/390 Supported Functions - Resource Specific Command Sets

And then proceed as for the previous two products for the installation. For more information on the Resource Specific Command Sets installation, please refer to Appendix D.3, “Installing MultiSystem manager” on page 249, and *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236.

5.2.1 Resource Specific Command Sets post-installation steps

Follow the instructions under “Modifying the Command Profile Editor Database” in the FLCREAD1.ME file and:

1. Open a Command Prompt window.

2. Move to the \usr\local\Tivoli\bin\w32-ix86\server\bin directory.
3. Issue the command: `CPEBATCH FLCI001N.RSP -I -G`
4. Issue the command: `CPEBATCH FLCD001N.RSP -I -G`

Note: The FLCD001N.RSP file is for IP discovery command sets, but was not documented in the read.me file at the time of writing.

5.2.2 NCM post-installation steps on Windows NT

After all three products are installed, you will see three icons on your Windows NT desktop as shown in Figure 130 on page 149.

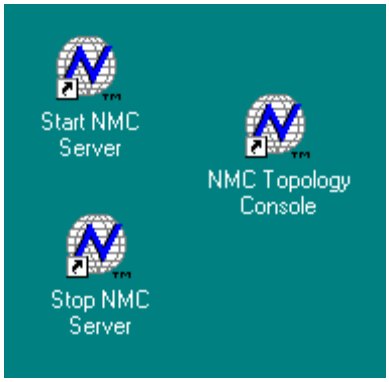


Figure 130. Tivoli NetView for OS/390 NMC icons

Now you can proceed to define the user rights for the NMC topology server to run in Windows NT. The user right that needs to be added to the user ID under which the NMC server will be run is:

- Act as part of the operating system.

Follow the instructions in the EGVREAD2.ME file (the read.me file for the NMC topology server) and then recycle your workstation.

5.3 NMC Startup

These are the steps needed in order to bring up all the required software on the workstation:

1. Start NMC topology server.
2. Issue the `NETCONV` command on the NetView for OS/390 NCCF command line to establish communications between NetView and NMC topology server.
3. Start the NMC topology console.

The following sections offer a complete description of each step.

5.3.1 Starting the NMC topology server

To start the NMC topology server, double click the NMC topology server icon on the Windows NT desktop or in the programs group under the Start button.

You will see a DOS box open on your desktop with these messages:

```
Tivoli environment variables configured.
BINDIR is set to j:\usr\local\Tivoli\bin\w32-ix86
IHS5903I: Starting NMC topology communications server.
IHS5904I: Starting NMC topology server.
Press any key to continue . . .
```

Figure 131. NMC topology server startup - messages in DOS window (1)

Press Enter, and you will see another DOS box opened with these messages:

```
....-9.24.106.77
....+9.24.106.7
--------------------------------------------------------------------
Tivoli NetView Management Console 1.3.0.0
Topology Communication Server
5697-B82 (C) Copyright Tivoli Systems, an IBM Company 1997, 1999.
All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Portions copyright (c) 1992 by Free Software Foundation, Inc. All rights
reserved
 --------------------------------------------------------------------
IHS0248I: Topology communications server 1.3.0.0 is initialized.
IHS0252I: Issue the 'tserver stop' command to end this process.
IHS0208E: The SNA communications server APPC subsystem is not loaded.
```

Figure 132. NMC topology server startup - messages in DOS window (2)

5.3.2 Starting communication with host

After NMC topology server has been initialized, issue the NETCONV command to set up communications between the NCM topology server and NetView for OS/390. The following command can be used on the NetView NCCF command line:

```
NETCONV ACTION=START, IP=x.x.x.x
```

where x.x.x.x is the IP address of the NMC topology server workstation.

Important

A new task DUIDGHB has been provided with the level set APAR OW41691. If you have this maintenance applied ensure that member DSIPARMU(DUIIGHB) contains the correct TCP/IP address space name as follows:

```
TCPANAME = TCPIPA
```

Check if task DUIDGHB has been started with the following command:

```
LIST STATUS=TASKS
```

If this task has not been started automatically by NetView, the command to activate it is:

```
START TASK=DUIDGHB,MOD=DSIZDST,MEM=DUIIGHB,PRI=5
```

The messages returned will be similar to those in Figure 133.

```
 NCCF                      Tivoli NetView    RA03N LUCY     03/09/00 11:11:09
- RA03N    DUI401I NETCONV COMMAND PROCESSED SUCCESSFULLY.  COMMUNICATION TO
             IP 23-8P4YK.itso.ral.ibm.com (9.24.106.77) STARTED.
 ------------------------------------------------------------------------------



???
NETCONV ACTION=START IP=9.24.104.77
```

Figure 133. NETCONV output

If the operator from which the command has been issued logs off from NetView, the connection with the topology server will end. To avoid this, you can issue the NETCONV command on an automated operator, such AUTO1, with this command:

```
EXCMD AUTO1,NETCONV ACTION=START IP=x.x.x.x
```

5.3.3 Starting the NMC topology console

To start the NMC topology console on Windows NT double click on the NMC topology console icon on the Windows NT desktop or in the programs group under the Start button. You will see the Sign On panel as shown in Figure 134.

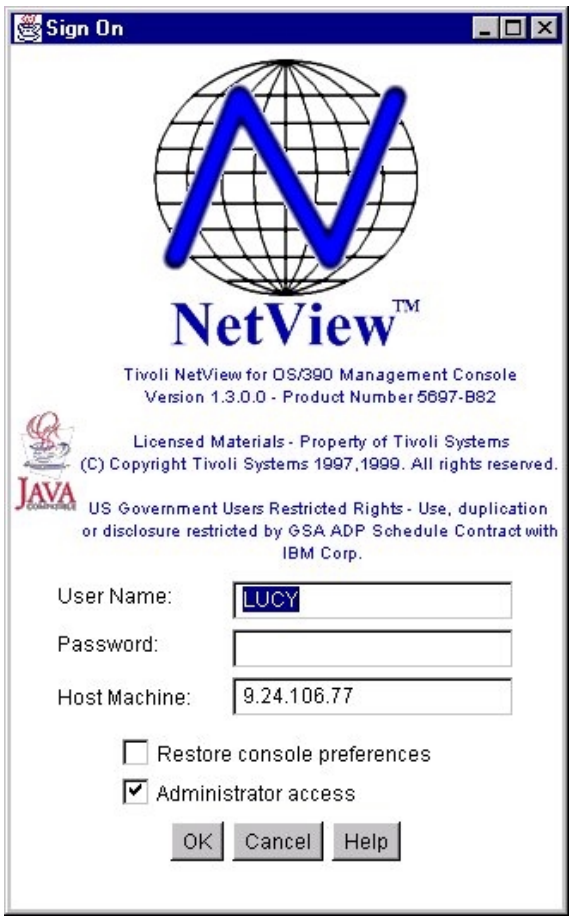


Figure 134. NMC topology console sign on panel

Then sign on to NMC:

Next enter the user name, password, and the IP address of the workstation where the NMC topology server is running. The user name must be a NetView operator ID that has actually logged on to NetView. Then click **OK**.

If you need to access NMC as an administrator (if you need to issue SNMP commands for example), be sure that `NGMFADMN=YES` has been configured in the profile definition for your NetView operator ID. See Table 27 on page 242 for further help on data set names.

All values except the password are saved and used as defaults the next time you sign on.

5.4 Operator Definitions

When operators are using the topology console, available commands are displayed in the context menu for a selected resource. Also, resource independent commands are shown in a context menu if you right click on the view background. Because operators have different areas of responsibility, levels of authority, and preferences, you may want to customize the command menus. The command profile editor enables you to control the content, order, and capabilities of these menus for an individual operator or group of operators.

NMC provides a command profile editor (CPE) GUI on Intel platforms and a batch utility on both Intel and UNIX platforms.

In our installation, the command profile editor GUI is stored in the C:\usr\local\Tivoli\bin\w32-ix86\TDS\server\bin directory, where “C” is the drive on which the NMC components have been installed.

By entering `cpe` in this directory, CPE is activated and you will see the CPE main window shown in Figure 135.

Figure 135. CPE main window

The new command sets for our operator Lucy need to be added. We proceeded with the following steps:

1. Open the Profile icon by double clicking it.
2. Click with the right mouse button on the default profile icon, then select **Copy**.

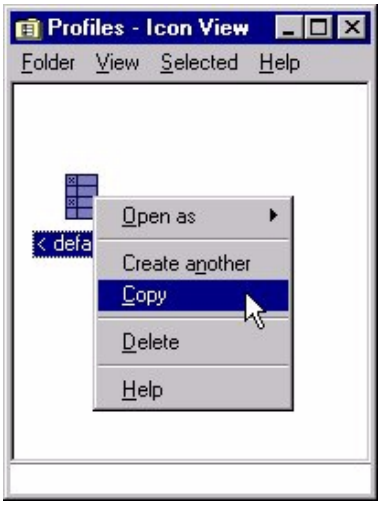


Figure 136. CPE: copy the default profile to create new profile

3. Type a Profile name and Comments.

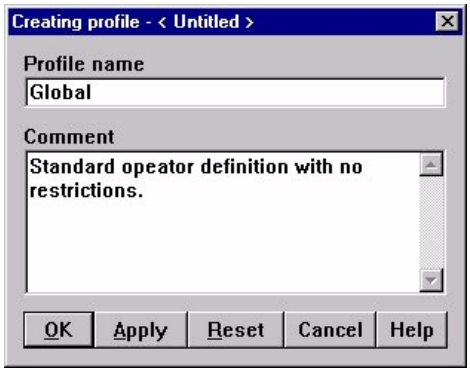


Figure 137. CPE: adding a new profile

4. Press **Apply** to apply your chosen settings. Press **OK** if you want to close this window.
5. Add a new operator by double-clicking the Operators icon, and select **Folder -> Add new operator** from the command menu.

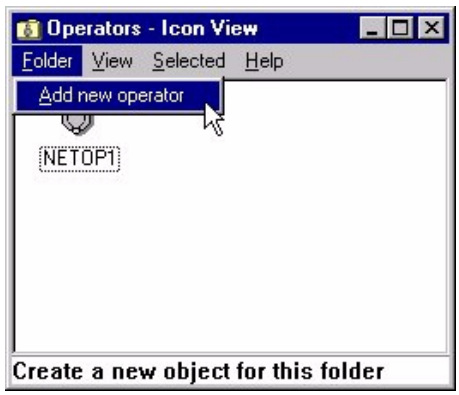


Figure 138. CPE: adding new operator

6. Now, put the operator name in the User name field (the same that you will use for the NMC console). This field is case-sensitive so it is better to use uppercase characters. Select a profile name from the drop-down list. Put any comments here if you need them and click the **OK** button.

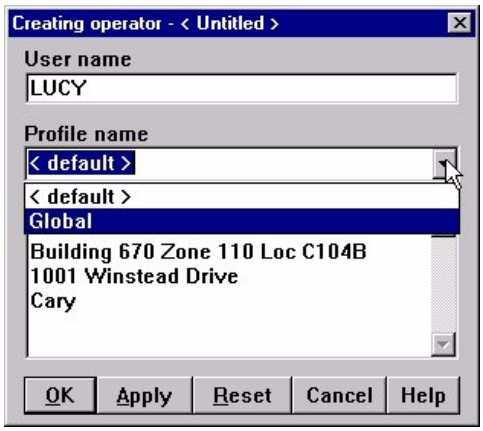


Figure 139. CPE: new operator definition

7. Now, save the new definitions by selecting **Save changes** in the main CPE window.

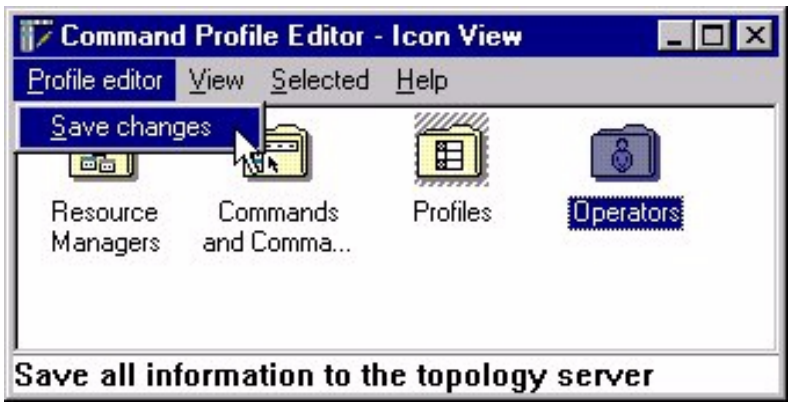


Figure 140. CPE: saving the configuration changes

8. Then, you can close the main CPE window

5.5 Using the NMC Main Window

The NMC main window consists of the following:

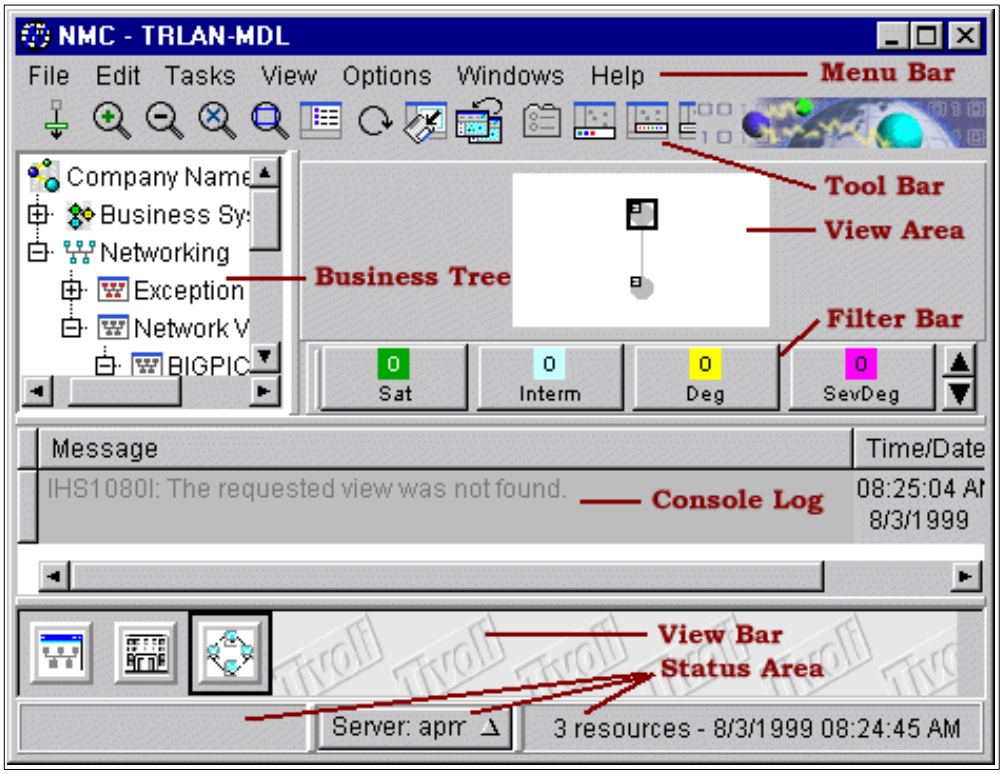


Figure 141. NetView Management Console main panel

- Menu bar: The menu bar contains pull-down menus, as shown in the diagram above.
- Tool bar: The tool bar contains icons that provide a quick way to perform certain functions found in the menu bar. Click here for more information about the icons.
- Business tree: The business tree lists the types of views that are available or currently open.
- View area: The view area displays the current topology or details view.
- Console Log: The console log displays messages and issued commands and their responses.
- Filter bar: The filter bar displays the number of resources in the current view that have each status or flag.
- View bar: The view bar displays all open views.
- Status area: The status area displays the following information:
  - Left: fly over text about the resource (that the mouse is over).
  - Center: status information, including NETCONV status and the name of the topology server to which you are connected.
  - Right: view information.

For more information on the usage of the NMC topology console, refer to *Tivoli NetView for OS/390 NetView Management Console User's Guide Version 1 Release 3*, GC31-8665.

5.6 Understanding resources

A real resource is a symbol on the screen that represents a physical or logical node or a physical or logical link. Real resources are mainly found in lower level views. An aggregate resource is a graphical representation of a collection of real resources. The status of an aggregate resource is based on the collective status and importance of the real links and nodes that make up the aggregate. Aggregate resources are found mainly in higher level views. Aggregate nodes are shown with the (+) symbol in the bottom-right corner. Aggregate links are shown with the (+) symbol near the link.

NMC uses geometric symbols, icons, text, colors, graphic modifiers (+, flags), and background pictures to identify resources in a network and show their statuses. The (+) modifier, shown in the lower-right portion of the resource, identifies aggregate resources. The flag modifier, shown in the upper-left portion of the resource, indicates that one or more important flags are set. Following is an example resource displaying the (+) modifier and an important flag set.

5.7 Java-based IP discovery

To populate RODM, Tivoli NetView provides the IP discovery function. The IP discovery function discovers and monitors IP resources such as subnets, endpoints, hosts, bridges and routers. It also does the following:

- Enables more information and control of SNMP-managed IP resources from the NMC.
- Discovers new, updated and deleted IP resources and stores them in RODM.
- Updates status dynamically, so there is no need for alerts or traps.

Note: Although Tivoli NetView MultiSystem Manager (MSM) and its IP agents are the primary way customers discover IP objects and get them into RODM and NMC, we will discuss only the IP collector functions provided by the Java-based IP discovery. At the time of writing, the Java-based IP discovery is a prototype and its implementation will likely be changed in the future releases. For more information about MSM and its IP agent, refer to *Tivoli NetView for OS/390 MultiSystem Manager User's Guide Version 1 Release 3*, GC31-8607.

5.7.1 Java-based IP discovery operational overview

Java-based IP discovery uses ICMP messages and SNMP polling to discover IP network topology. Ping is used to detect if an interface is up or down and SNMP polling is used to query the system information. System information includes how many interfaces are in the system, whether the system is a router or bridge and the network which the system is in along with which router each interface is connected to. A subset of the information is loaded into RODM.

IP discovery also performs special processing for unnumbered IP interfaces, HSRP (Hot Standby Router Protocol), interfaces with secondary IP addresses, and DHCP (Dynamic Host Configuration Protocol).

IP discovery begins discovering from the IP address of the OS/390 TCP/IP stack it is running on and IP addresses provided as seeds in the configuration file /etc/netview/ipdiscovery.conf (see Appendix E.9.3, "/etc/netview/ipdiscovery.conf"

on page 309 for our implementation). New addresses are discovered from the ARP tables of discovered systems. If discovered addresses are not within the scope of discovery configured, they will be ignored. To determine if addresses are within the scope, an IP mask is applied. If the IP mask of the system is not yet known (not SNMP polled yet), the network IP mask is used. If there is a match, SNMP polling of the system's ARP table is performed and the system's IP mask is then determined. For each successive IP address discovered within the scope of discovery, if the address cannot be pinged and is not a seed address, it is ignored. If it is a seed address, IP discovery will attempt to ping it again at intervals thereafter. If a system with an IP address within the scope of discovery is up, that means the interface is reachable using ping, system information is queried using SNMP.

IP discovery sends the SNMP GET requests against the sysName, sysOid, sysDescr, sysContact and sysLocation MIB variables. The ipForwarding MIB is also queried to determine if the system is a router. RFC 1493 STP MIB is checked to determine if the system is a bridge. Finally, ipAddrTable and ifTable are queried to determine the number of interfaces in the system.

If a discovered system within the scope of discovery doesn't have a valid SNMP agent, NetView IP discovery will still create objects representing the computer system with a single interface. If the IP address has been discovered from an ARP table, the subnet mask should be known. If not, the subnet mask will be determined if the address is found in a subsequently discovered system's ARP table. The mask determines the network the node is in. If the network hasn't been discovered before and the address is a seed address, IP discovery creates a new object for the network and then the new system, within that network. If the node is not a seed address and the applicable network object for this address hasn't been created yet, it is assumed that the SNMP information for this system is incorrect and the system is then ignored.

If the system is a router, every interface will be polled. For each interface in a network not yet created in RODM, a network object is created first.

The status of an interface object is set to OK if the ping is successful; otherwise it is set to Error. If the SNMP poll of the system determines that ifAdminStatus is down, the status is set to Disabled.

5.7.2 Java-based IP discovery installation

This section describes the installation and configuration of Java-based IP discovery.

5.7.2.1 Prerequisite components

NetView IP discovery has the following prerequisites:

- The Tivoli NetView UNIX command server must be installed and operational. For information on the UNIX command server installation, consult Appendix D.4.1, “How to start the UNIX server” on page 253.
- Java for OS/390 at the Java Development Kit (JDK) 1.1.6 level. The IP discovery sample code has been developed using the OS/390 JDK Version 1.1.6. In our installation we used Java for OS/390 Version 1.1.8.
- The Java-based IP discovery sample code must be installed. The function is in two parts. First the IP discovery engine which is a self-contained Java

application running under UNIX System Services must be present. The second part of the function exists within NetView to start, stop and receive data from IP discovery. Data received from IP discovery is used to update RODM.

- Tivoli NetView MultiSystem Manager. You must also have the Resource Specific Command Sets installed.

5.7.2.2 NetView UNIX command server configuration

The following changes must be made to a customer's implementation of the NetView UNIX command server JCL in DSIPARMU(CNMSJUNX):

- Add the Java and NetView/390 (already included in shipped sample) executable directories to the PATH statement.
- Add the RESOLVER_CONFIG environment variable assignment assigning it the value of your TCP/IP configuration file.
- Add CLASSPATH statement to include ihssnmps.jar, ipdiscovery.jar and classes.zip for Java.
- Add LIBPATH and LD_LIBRARY_PATH environment variables assigned to the NetView /lib directory.

Additionally, it is recommended that REGION=0M be added on the CNMSJUNX JOB card, if the UNIX Command Server is being run as an MVS batch job.

See Appendix D, “NetView sample configuration” on page 241 for the complete example in our configuration.

5.7.2.3 MultiSystem Manager configuration

Once the MultiSystem Manager host code has been configured, perform the following to enable the IP discovery sample:

1. Define a new autotask for IP discovery. The default is AUTODIS1. Define this user ID to RACF with an OMVS segment with UID(0).
2. Modify the DSIPARMU(FLCAINP) member by uncommenting and modifying the following line:
   - START_DISCOVERY, which is used to start the IP discovery agent.
   - The JAVA_PATH, JAR_PATH, and DCONF_PATH keyword to meet your installation.

Figure 142 shows our MultiSystem Manager initialization parameters configured in DSIPARMU(FLCAINP):

```
START_DISCOVERY=AUTOTASK=AUTODIS1 GATHER_OBJECTS=1000 GATHER_TIME=100

JAVA_PATH=/usr/lpp/java/J1.1/lib/classes.zip
JAR_PATH=/usr/lpp/netview/lib/ipdiscovery.jar
DCONF_PATH=/etc/netview/ipdiscovery.conf
```

Figure 142. Sample MultiSystem Manager configuration - DSIPARMU(FLCAINP)

See Appendix D, “NetView sample configuration” on page 241 for the complete implementation at ITSO Raleigh. Further information on the MSM configuration can be found in *Tivoli NetView for OS/390 Installation and Administration Guide*

Version 1 Release 3, SC31-8236, and *Tivoli NetView for OS/390 MultiSystem Manager User's Guide Version 1 Release 3*, GC31-8607.

5.7.3 Java-based IP discovery sample configuration

The Java-based IP discovery sample is installed as part of the SMP/E installation of Tivoli NetView for OS/390 V1R3. In most installations, the HFS files related to the NetView function are installed in the /usr/lpp/netview directory, where the NetView HFS data set is mounted.

The IP discovery function uses the following two configuration files:

- ipdiscovery.conf: This file is used primarily to specify seed IP addresses for discovery to start from and the scope of addresses to limit discovery. The log file name also can be specified in this file.
- snmp.conf: IP discovery uses this configuration file to obtain various parameters needed to send the SNMP requests, such as an SNMP community name and retry timer value. The timeout intervals and retry counts for ping operations also can be defined in this file.

The sample files can be found in /usr/lpp/netview and have to be copied to the /etc/netview directory.

These two configuration files have been updated to meet our environment requirements as shown in Figure 143 and Figure 144.

```
# dumpFile=/quanta/data/ipd.dump
logFile=/tmp/discovery.log                1

snmpFile=/etc/netview/snmp.conf           2

scope=*.*.*.*

seeds=9.24.104.113,9.24.104.149           3

debugMask=snmp,ping,discovery,topology
```

Figure 143. /etc/netview/ipdiscovery.conf sample configuration

1 All log messages from the IP discovery agent are written in this log file.

2 This is the name of the file in which SNMP parameters have been defined.

3 This is the seed definition. IP discovery will begin discovery with these IP addresses.

```
9.24.104.113:3s:3:30m:publicv1:5s:1:1d:publicv1:161:*    1
9.24.104.42:3s:3:30m:publicv1:5s:1:1d:publicv1:161:*
9.24.104.149:3s:3:30m:publiv1:5s:1:1d:publicv1:161:*
9.24.105.126:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
9.24.105.73:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.250.3:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.252.28:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.232.39:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
9.24.105.*:3s:3:30m:public:2s:1:1d:public:161:*
*.*.*.*:3s:3:30m:public:2s:1:1d:public:161:*              2
```

Figure 144. /etc/netview/snmp.conf sample configuration

1 Define the associations between a community name and an IP address, or IP subnet. One or more IP addresses and IP subnets can be specified separated by commas.

2 This is the default entry, which is used for IP addresses that do not match the entries configured in this configuration file.

Important

At the time of writing, while the IP discovery agent supports multiple community names separated by commas in one entry in the snmp.conf file, the Java SNMP command doesn't. Therefore if you are planning to use the MIB browser, do not specify more than one community name in one field.

Figure 145 shows the sample log messages from the IP discovery agent. You will see various information related to the discovery, such as the community name used for a particular IP host **1**, the result of a ping attempt **2**, and MIB variables requested by the IP discovery agent **3**.

```
==> Send ping 1/3 to : 9.24.106.188
<== Recevied ping response from : 9.24.106.188            2
==> Send ping 1/3 to : 9.24.104.26
<== Recevied ping response from : 9.24.104.26
==> Try community name publicv1 to 9.24.104.113           1
==> Found community name publicv1 from 9.24.104.113
==> Send snmp com.tivoli.net.discovery.ipdiscovery.SnmpQuery@2ce43f0 to 9.24.104.1
<== Recevied snmp response com.tivoli.net.discovery.ipdiscovery.SnmpQuery@2ce43f0
9.24.104.113
==> Try community name public to 9.24.104.100
==> Found community name public from 9.24.104.100
==> Send snmp ipNetToMediaTable to 9.24.104.100
<== Recevied snmp response ipNetToMediaTable from 9.24.104.100
IpNetToMediaTable: ifIndex 1 netAddr 9.24.106.100 physAddr 40002216AA00  3
```

Figure 145. Sample log message from the IP discovery agent

For further information on the IP discovery installation and configuration, refer to Appendix D.6, "Installing Java-based IP discovery agent" on page 259. The complete descriptions on the IP discovery configuration can be found in *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236.

5.8 MIB browser

The MIB browser now communicates with the Java SNMP service components on OS/390 and provides interactive MIB information. The SNMP engine communicates with the MIB metadata server to provide easy to understand MIB information and to obtain data from the SNMP agents. Network administrators can display MIB variables in Java-based GUI panels, and update their values.

The MIB browser requires that the Java SNMP services be configured. The SNMP manager shipped with CS for OS/390 IP (osnmp) cannot be used to retrieve MIB variables for the MIB browser.

5.8.1 Enabling MIB browser

Before you use the MIB browser from the NMC console the following tasks have to be completed:

1. Start the Java SNMP services (see Appendix D.5, “Install the Java SNMP services” on page 255).
2. Copy configuration files to NMC servers.

   Copy the updated file /etc/netview/nv390srvr.conf (see Figure 201 on page 258) from UNIX System Services to your NMC server workstation. Copy the file to the following directory and use the following file name:

   ```
   /usr/local/Tivoli/bin/w32-ix86/TDS/server/db/current/settings/defaultmibbr.properties
   ```

   Note: If you change /etc/netview/nv390srvr.conf, repeat this step so that the proper configuration is reflected at your NMC servers.
3. Activate MultiSystem Manager (see Appendix D.7.1, “Initialization operations” on page 263 for the MSM startup).

The MIB browser can be launched for SNMP-managed resources whose IP address has been discovered. To have MultiSystem Manager resources displayed by NMC, MultiSystem Manager must be loaded and running.

5.8.2 Launching the MIB browser

You can launch the MIB browser window from the topology console screen (see Figure 146).

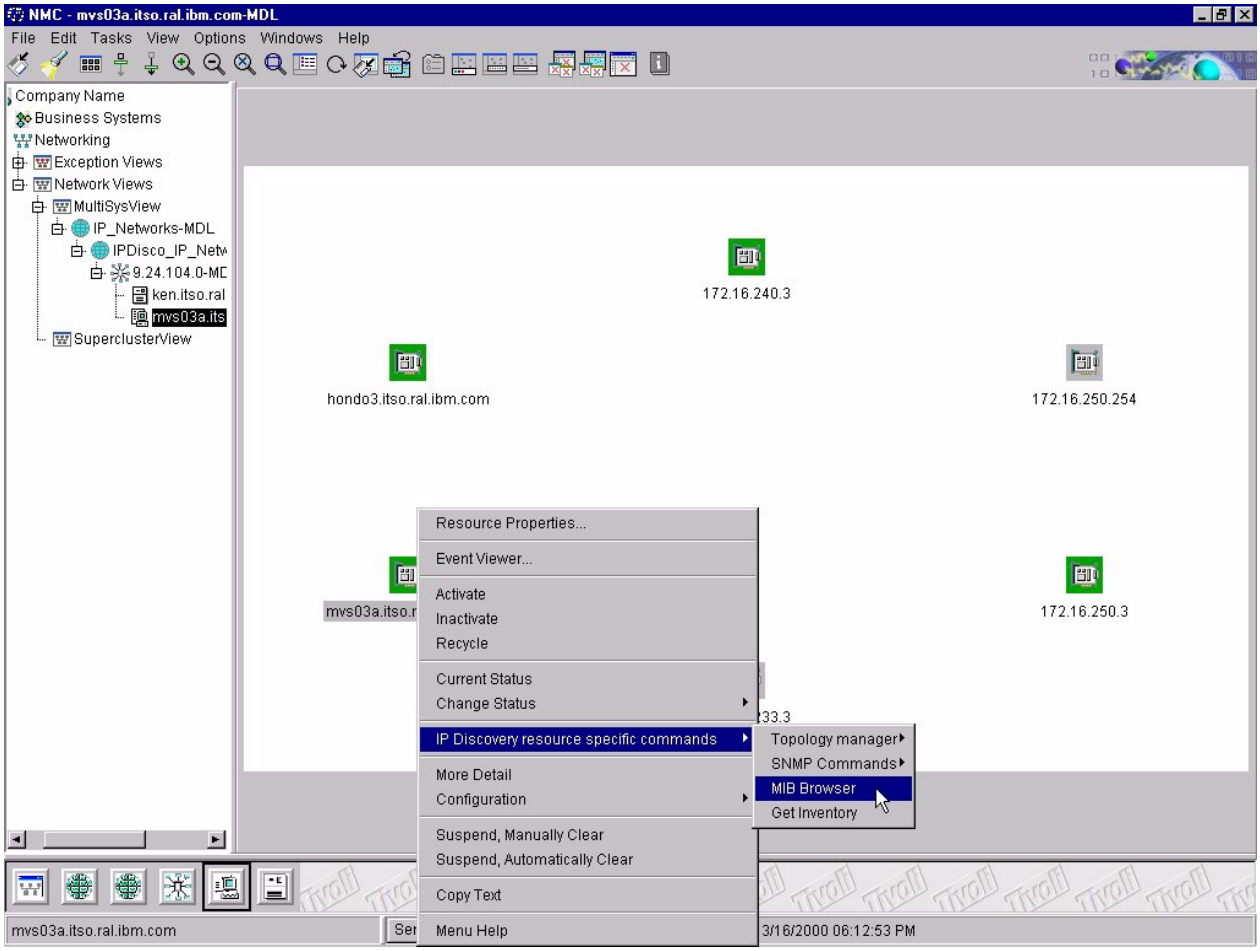


Figure 146. NMC window - select MIB browser

If you right click on one resource, then select **IP Discovery resource specific commands -> MIB Browser** you will see a screen displayed similar to the one shown in Figure 147 on page 163.

Figure 147. MIB browser window

You can display a particular MIB variable on the MIB browser. It is also possible to walk through the MIB variables. In the sample output shown in Figure 148 on page 164, we displayed all MIB variables contained in the ibmTcpipMvsDeviceTable MIB object.

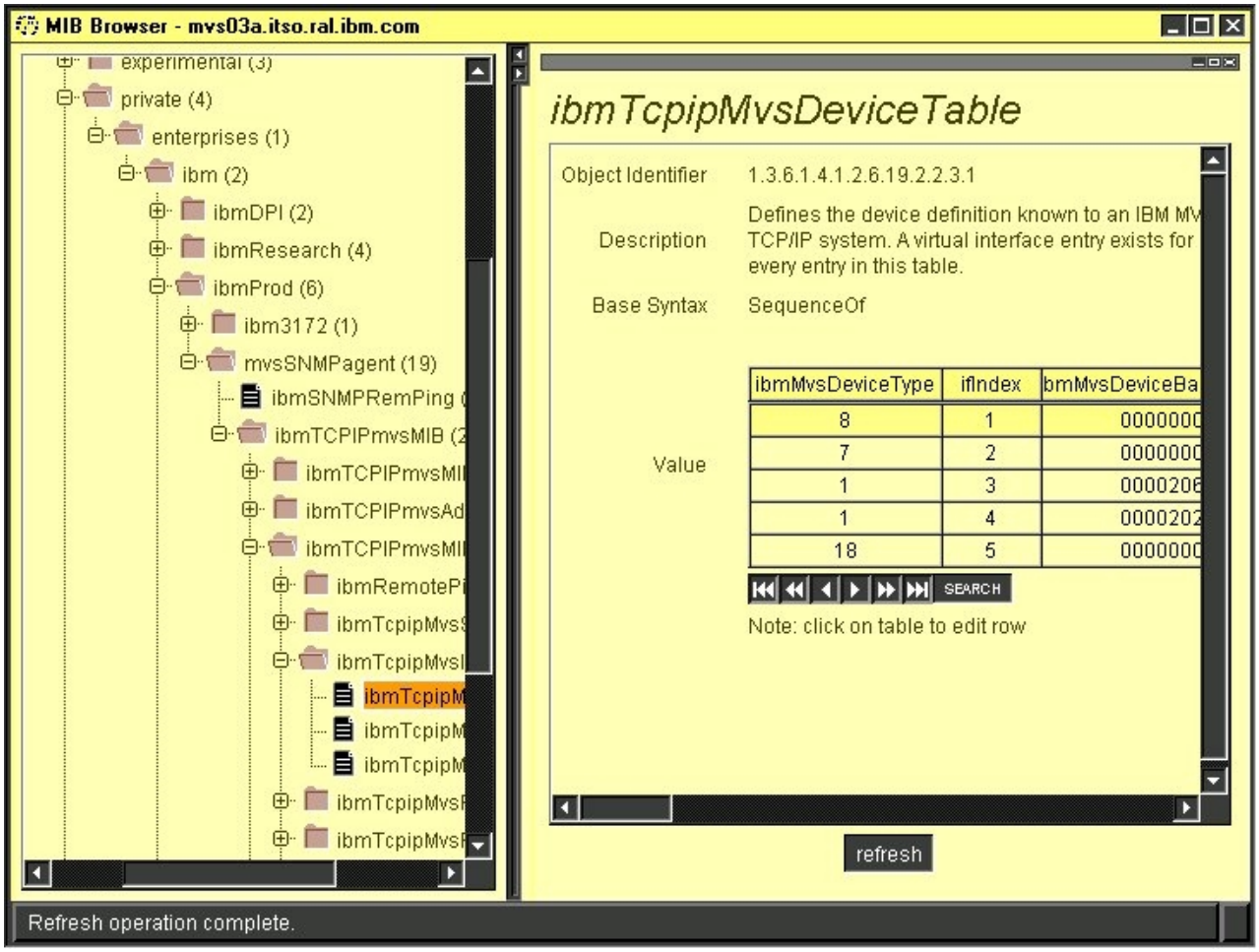


Figure 148. Display MIB variables

If you want to define security authorization for the MIB browser, you can create NetView span-of-control security authorization tables using command list CNMEMIBR (read) and CNMEMIBW (read/write). For more information about the MIB browser security, refer to *Tivoli NetView for OS/390 Security Reference Version 1 Release 3*, SC31-8606.

Appendix A. Working with the NetView SNMP command

In this appendix, the configuraiton of the NetView SNMP command which was introduced in IBM TCP/IP for MVS V2R2.1 and provides the SNMP manager functions will be discussed. Note that, the NetView SNMP command supports SNMPv1 only and runs in the NetView environment.

A.1 Implementation overview

The SNMP command in the NetView environment can be used to send SNMP Version 1 requests to SNMP agents on either local or remote hosts. The SNMP command requires the command processor itself, the SNMPIUCV task for inter-address space communication, and the SNMP query engine (SNMPQE), which creates the packets sent to the SNMP agent. The NetView SNMP command supports only community-based security.

Figure 149 illustrates the implementation of the SNMP NetView command in CS for OS/390 IP.

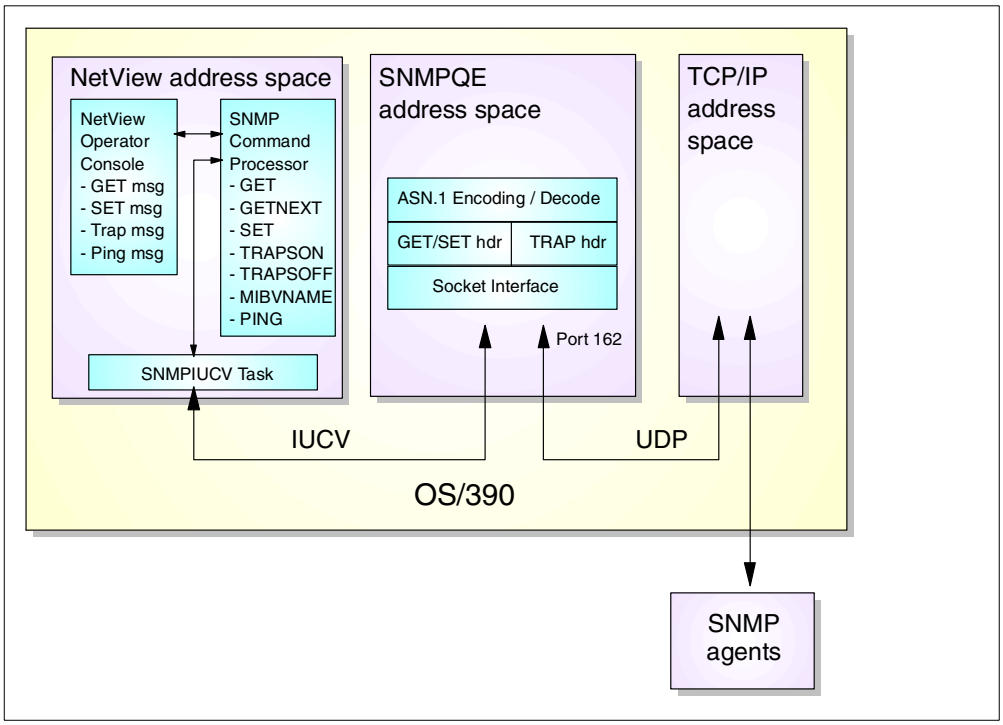


Figure 149. NetView SNMP command implementation

The following two address spaces are required to run the NetView SNMP command:

- NetView with SNMP network management station support. NetView Version 1 Release 3 or later is required for the SNMP monitor function.
- SNMPQE - SNMP query engine, which is an address space that works as an interface between NetView and TCP/IP.

The IUCV that SNMPQE uses is the IUCV/VMCF interaddress space communication service that enables applications running in the same OS/390 image to communicate with each other without requiring the services of the TCP/IP protocol stack. Therefore you must have the VMCF and TNF subsystems configured and running.

Note that the IUCV/VMCF interaddress space communication service is different than the IUCV socket API, which is not supported in CS for OS/390.

A.2 Configure the NetView SNMP command

In this section, we will describe how to configure the NetView SNMP function. The NetView SNMP command requires the command processor itself, the SNMPIUCV task for inter-address space communication, and SNMPQE, which creates the packets sent to the SNMP agent.

A.2.1 Configure the SNMP query engine

The SNMP query engine runs in the SNMPQE address space. You have to update the sample procedure (SNMPPROC) given in TCPIP.SEZAINST to suit your local conditions and your naming standards.

SNMPQE needs access to the *hlq*.MIBDESC.DATA data set for the MIB variable descriptions. The sample of this file is provided in TCPIP.SEZAINST(MIBDESC). In most installations, the MIBDESC.DATA data set can be found in TCPIP.MIBDESC.DATA, where *hlq* is set to TCPIP.

The MIBDESC.DATA data set defines the short names for MIB variables. Short names are the character representation for the ASN.1 variable names. For example, sysUpTime is the short name for 1.3.6.1.2.1.1.3.0 (the MIB variable that stores the time since the SNMP agent was last restarted). Short names are generally shown as a combination of upper and lowercase characters, though SNMP on CS for OS/390 ignores these case distinctions. Variable names must always be in ASN.1 language when they are sent to an SNMP agent. You can always use ASN.1 language to specify the variable names in an enterprise-specific tree (assuming that the agent supports them). You can use these short names to specify the MIB variables.

Figure 150 is the sample started procedure we used for the SNMP query engine:

```
//SNMPQE   PROC MODULE=SQESERV,PARMS=''
//*SNMPQE   PROC MODULE=SQESERV,PARMS='-d 4'
//*
//SNMPQE   EXEC PGM=&MODULE,PARM='&PARMS',
//             REGION=4096K,TIME=1440
//*
//STEPLIB  DD DSN=TCPIP.SEZADSIL,DISP=SHR
//SYSPRINT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//SYSIN    DD DUMMY
//SYSMDUMP DD SYSOUT=*
//*MSSNMPMS DD DSN=TCPIP.SEZAINST(MSSNMP),DISP=SHR
//*
//SYSTCPD  DD DSN=TCPIP.TCPPARMS(TCPD&SYSCLONE.A),DISP=SHR
```

Figure 150. SNMP query engine sample started procedure

Since the NetView SNMP command uses RAW sockets to issue the SNMP PING requests, the user ID associated with the SNMPQE started task needs to have superuser authority in OS/390 UNIX environment, that is OS/390 UNIX UID of zero.

A.2.2 Configure NetView as an SNMP monitor

To configure the NetView interface as an SNMP monitor, perform each of the following tasks:

1. Add SNMPIUCV task to NetView.
2. Add SNMP initialization parameters.
3. Add the `SNMP` command to NetView.
4. Add SNMP message to NetView.

A.2.2.1 Add SNMPIUCV task to NetView

The SNMPIUCV is the task that handles the communication between NetView and SNMPQE. Add the following line to the DSIDMN member of the DSIPARM data set:

```
TASK     MOD=SNMPIUCV,TSKID=SNMPIUCV,PRI=5,INIT=Y
```

This statement causes the task SNMPIUCV to start automatically when NetView is started.

If you specify INIT=N instead of INIT=Y in the TASK statement for SNMPIUCV, a NetView operator can start the SNMPIUCV task by entering the following:

```
START TASK=SNMPIUCV
```

The SNMPIUCV task tries to connect through IUCV to SNMPQE. If this fails, it retries the connection as specified by the SNMPQERT keyword in the SNMPARMS member of the hlq.SEZADSIP data set. The default is every 60 seconds.

The SNMPIUCV module resides in the TCPIP.SEZADSIL data set. This data set should be concatenated to the SETPLIB DD statement in the NetView start procedure.

A.2.2.2 Add SNMP initialization parameters

The SNMPIUCV task reads a member named SNMPARMS. This member is in the data set TCPIP.SEZADSIP. You may add this data set to DSIPARMS into an existing data set in the DSIPARM DD statement in the NetView start procedure.

If you change the name of the address space of the SNMP query engine, you must change the SNMPQE parameter in SNMPARMS.

```
SNMPQE   SNMPQE  1  * Userid of SNMP Query Engine
SNMPQERT 60         * Retry timer (seconds) for IUCV CONNECT
SNMPRCNT 2          * Retry count for sending SNMP requests
SNMPRITO 10         * Retry initial timeout (10ths of a second)
SNMPRETO 2          * Retry backoff exponent (1=linear, 2=exponential)
SNMPMMLL 80         * Line length for Multiline Messages 38/44
```

Figure 151. TCPIP.SEZADSIP(SNMPARMS) configuration

1 is the name of the SNMP query engine started procedure. The default address space name is SNMPQE.

Refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513, for a full description of the parameters of the SNMPARMS member.

A.2.2.3 Add the SNMP command to NetView

The SNMP command processor that allows NetView operators and CLISTs to issue SNMP commands must be defined in the DSICMD member of the data set specified by the DSIPARM DD statement in the NetView start procedure as follows:

```
SNMP      CMDMDL MOD=SNMP,ECHO=Y,TYPE=R,RES=Y
```

By coding `RES=Y`, you can make the command processor resident.

After the SNMPIUCV task is started, you can issue the SNMP command. The SNMP command passes a request to the SNMPIUCV task to forward it to SNMPQE. The return code represents a request number that is associated with the request. The responses are returned asynchronously and contain this request number. The operator or CLIST must use the request number to correlate the response to the request.

The SNMP module resides in the TCPIP.SEZADSIL data set. This data set should be concatenated to the SETPLIB DD statement in the NetView start procedure.

A.2.2.4 Add SNMP message to NetView

The SNMP messages are in hlq.SEZADSIM, where hlq is TCPIP in most installations. The valid message members are DSISNM00 through DSISNM05, DSISNM10, DSISNM12, and DSISNM99. This data set should be concatenated to the DSIMSG DD statement of the NetView start procedure.

Table 24 summarizes the NetView configuration that is required in most installations.

Table 24. Summary of the NetView customization

Customization for	Typical customization
STEPLIB DD statement	Concatenate the TCPIP.SEZADSIL data set
SNMPIUCV task definition	Update DSIPARM(DSIDMN) member
SNMP command processor	Update DSIPARM(DSICMD) member
SNMP messages	Concatenate the TCPIP.SEZADSIM data set to the DSIMSG DD statement

A.2.3 Getting started

When you have configured these required settings, you can start NetView and the SNMP query engine.

The SNMPIUCV task tries to connect to SNMPQE using IUCV, when it is started. If it fails, it retries the connection. When the SNMPIUCV task fails to comment to SNMPQE, the warning message below will be shown in the MVS console periodically until it successfully establishes an association with SNMPQE:

```
SNM101W SNMP task (SNMPIUCV) found Query Engine (SNMPQE) not ready
```

When the SNMPIUCV task has established the connection successfully, the following messages will be shown at the MVS console:

```
SNM100I SNMP task (SNMPIUCV) connecting to Query Engine (SNMPQE) path=1
SNM102I SNMP task (SNMPIUCV) connected to Query Engine (SNMPQE), path=1, msglim=1
```

A.3 Working with the NetView SNMP command

In this section, we will show several examples where we used the SNMP command processor under NetView. Table 25 summarizes the SNMP operations supported by the NetView SNMP command.

Table 25. NetView SNMP request types

SNMP request type	Description
Get	Sends a GetRequest-PDU to an SNMP agent for a specific MIB object value.
GETNext	Sends a GetNextRequest-PDU to an SNMP agent for the MIB object value that lexicographically follows the MIB object specified.
Set	Send a SetRequest-PDU to an SNMP agent to set a specific MIB variable.
PING	Send an ICMP echo request message and obtain the minimum round-trip response time from SNMPQE to a specific node.
MIBvname	Displays the textual name of an ASN.1 MIB object.
TRAPson	Request SNMPQE to receive SNMP traps destined to OS/390, and display them on the NetView panel.
TRAPSOFf	Request SNMPQE to stop listening on port 162 for SNMP traps.

A.3.1 Issuing the SNMP commands to maintain MIB objects

Figure 152 shows an example of the SNMP GET command to retrieve MIB values maintained by SNMP agents.

```
* RA28N    snmp get 9.24.104.149 publicv1 sysDescr.0
+ RA28N    SNM050I SNMP Request 1001 from KAKKY accepted, sent to Query Engine
= RA28N
SNM040I SNMP Request 1001 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.1.0       2
SNM043I Variable value type: 9
SNM044I Variable value: SNMPv3 agent version 1.0 with DPI version 2.0   3
SNM049I SNMP Request 1001 End of response
--------------------------------------------------------------------------------




netvasis snmp get 9.24.104.149 publicv1 sysDescr.0     1
  2                                          3
```

Figure 152. NetView SNMP GET command

1 is the command issued from NetView NCCF panel.

When you enter a command from the NetView Command Facility command line, the NetView program converts lowercase characters to uppercase prior to processing. Prefixing your commands with NETVASIS prevents this conversion and allows you to enter commands in mixed case.

Since the community name is case-sensitive, the NETVASIS command (2) needs to be issued with the SNMP command.

When the SNMP command is issued from a CLIST, input is passed in whatever case it was passed from the CLIST, which means mixed case characters can be used.

The textual names for the variables (3) passed to the query engine are compared against the entries in the MIBDESC.DATA file. This comparison is not case sensitive.

Messages SNM042I - SNM049I return the answer from SNMPQE. 2 shows the MIB object identifier, which corresponds to sysDescr. 3 shows the current value of this MIB object.

```
* RA28N    snmp getnext 9.24.104.149 publicv1 sysContact.0
+ RA28N    SNM050I SNMP Request 1007 from KAKKY accepted, sent to Query Engine
= RA28N
SNM040I SNMP Request 1007 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.5.0                  3
SNM043I Variable value type: 9
SNM044I Variable value: CS for OS/390 V2R8 on SYSTEM39     2
SNM049I SNMP Request 1007 End of response
--------------------------------------------------------------------------------



 ???
netvasis snmp getnext 9.24.104.149 publicv1 sysContact.0
                                            1
```

Figure 153. NetView SNMP GETNEXT command

The NetView SNMP command also supports the SNMP GETNEXT operation which sends a request to an SNMP agent for the next MIB variable that lexicographically follows the object name specified in the command.

In the example shown in Figure 153, the sysContact.0 object (1) is given to the command. Then as an output, the value of the sysName.0 (2) which is defined after sysContact.0 in the system group is returned.

“1.3.6.1.2.1.1.5” (3) is the object identifier of sysName, and sysContact has the object identifier of “1.3.6.1.2.1.1.4”.

```
SNM049I SNMP Request 1009 End of response
* RA28N    snmp set 9.24.104.149 publicv1 sysContact.0 'Lucia Santucci'
+ RA28N    SNM050I SNMP Request 1010 from KAKKY accepted, sent to Query Engine
= RA28N
SNM040I SNMP Request 1010 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.4.0
SNM043I Variable value type: 9
SNM044I Variable value: Lucia Santucci
SNM049I SNMP Request 1010 End of response
--------------------------------------------------------------------------------




???
netvasis snmp set 9.24.104.149 publicv1 sysContact.0 'Lucia Santucci'
             1
```

Figure 154. NetView SNMP SET command

The SNMP SET request (1) can be sent by the NetView SNMP command. In the example, Figure 154, we change the name of the contact person for this system.

```
= RA28N
SNM040I SNMP Request 1054 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.4.0         a
SNM043I Variable value type: 9
SNM044I Variable value: Tatsuhiko Kakimoto
SNM042I Variable name: 1.3.6.1.2.1.1.1.0         b
SNM043I Variable value type: 9
SNM044I Variable value: SNMPv3 agent version 1.0 with DPI version 2.0
SNM042I Variable name: 1.3.6.1.2.1.1.6.0         c
SNM043I Variable value type: 9
SNM044I Variable value: ITSO Raleigh, IBM
SNM049I SNMP Request 1054 End of response
--------------------------------------------------------------------------------

???
netvasis snmp get 9.24.104.149 publicv1 sysContact.0 sysDescr.0 sysLocation.0
                                        1          2         3
```

Figure 155. SNMP command with multiple MIB objects specified

Multiple variables can be specified with the GET, GETNext, or SET commands, which are all packaged in one SNMP PDU to be sent to the SNMP agent. See Figure 155 for an example.

1, 2, and 3 are the MIB object names specified for the SNMP GET command, and a, b, and c are the corresponding values for these MIB objects returned from the SNMP agent.

```
 * RA28N    SNMP PING 9.170.3.1
 + RA28N    SNM050I SNMP Request 1012 from KAKKY accepted, sent to Query Engine
 = RA28N
 SNM040I SNMP Request 1012 from KAKKY Returned the following response:
 SNM042I Variable name: 1.3.6.1.4.1.2.2.1.3.2.9.170.3.1    3
 SNM043I Variable value type: 1
 SNM044I Variable value: 276        2
 SNM049I SNMP Request 1012 End of response
 ------------------------------------------------------------------------------




 ???
SNMP PING 9.170.3.1
            1
```

Figure 156. NetView SNMP PING command

Using the NetView SNMP PING command (1) you can verify the reachability to the destination IP host, and obtain the round trip time to it (see Figure 156).

When the SNMP PING command is issued, SNMPQE sends one PING, which is an ICMP echo on a raw socket, and returns the value in milliseconds (2) in an IBM-defined SNMP variable minRTT (3), which is suffixed with the destination IP address.

If the PING does not respond, SNMPQE retries twice, once after one second and again after two seconds (SNMPQE default retry mechanism). If a response is not received after all retries have been exhausted, a variable value of -1 is returned to indicate that a reply was not received.

SNMP uses ICMP Echo to send a PING command to the remote host. No SNMP PDU exchange with the remote host occurs. Therefore, a successful SNMP PING indicates only that the remote host is active and reachable. It does not indicate that the SNMP agent at the remote host is active, or that the SNMP manager can send requests to the SNMP agent if it is active.

```
* RA28N    SNMP MIBVNAME 1.3.6.1.2.1.1.1
+ RA28N    SNM050I SNMP Request 1010 from KAKKY accepted, sent to Query Engine
= RA28N
SNM040I SNMP Request 1010 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.1.1
SNM043I Variable value type: 9
SNM044I Variable value: sysDescr                                    2
SNM049I SNMP Request 1010 End of response
--------------------------------------------------------------------------------


???
snmp MIBvname 1.3.6.1.2.1.1.1
        1
```

Figure 157. NetView SNMP MIBVNAME command

You can use the NetView SNMP MIBVNAME command (1) to find the textual MIB object name from the corresponding MIB object identifier. The response arrives in the NetView program asynchronously, and then it displays the response as a multiline message (2) as shown in Table 157.

A.3.2 Monitoring traps in the NetView environment

SNMP trap messages can be captured by SNMPQE and shown in the NetView NCCF panel. To enable SNMPQE to forward SNMP traps to NetView, you must first set the required trap filters using the SNMP TRAPSON command.

```
NCCF                    N E T V I E W    RA28N KAKKY    03/07/00 14:39:06 A
* RA28N    SNMP TRAPSON 255.0.0.0 9.0.0.0     1
+ RA28N    SNM050I SNMP Request 1028 from KAKKY accepted, sent to Query Engine
= RA28N               2
SNM040I SNMP Request 1028 from KAKKY Returned the following response:
SNM045I Major error code: 0
SNM046I Minor error code: 0
SNM047I Error index: 0
SNM048I Error text: no error
SNM049I SNMP Request 1028 End of response
--------------------------------------------------------------------------------



???
snmp trapson 255.0.0.0 9.0.0.0      1
```

Figure 158. Enabling NetView to receive and forward SNMP traps

With the TRAPSON command 1, we request SNMPQE to forward all traps from IP hosts belonging to the 9.0.0.0 network. The trap filter ID returned to NetView from the SNMPQE address space in this example is 1028 (2).

If we only wanted to receive traps from the 9.24.104 subnet, we would have entered a command of the following form:

```
SNMP TRAPSON 255.255.255.0 9.24.104.0
```

The first operand (255.255.255.0) is the mask, which the query engine uses to *and* with the origination IP address of a trap. If the results of the operation equals the value of the second parameter (9.24.104.0), the trap will be forwarded to NetView.

```
 * RA28N    SNMP TRAPSOFF 1028     3
 + RA28N    SNM050I SNMP Request 1030 from KAKKY accepted, sent to Query Engine
 = RA28N
 SNM040I SNMP Request 1030 from KAKKY Returned the following response:
 SNM045I Major error code: 0
 SNM046I Minor error code: 0
 SNM047I Error index: 0
 SNM048I Error text: no error
 SNM049I SNMP Request 1030 End of response
 ------------------------------------------------------------------------------



 ???
snmp trapsoff 1028      3
               4
```

A trap filter remains in effect until SNMPQE is recycled or it is removed explicitly by issuing the TRAPSOFF command 3. The trap filter ID (4), which was passed back to you as part of the response to the TRAPSON command, has to be specified with the TRAPSOFF command. The trap filter ID is also displayed in the header message of traps passed by this filter.

A trap message forwarded by SNMPQE is sent to the NetView operator who is designated as the authorized receiver defined in the operator profile. Therefore traps may be shown on the console of a different operator than the one who issued the TRAPSON command.

We coded the following parameters for the NetView operator profile to display SNMP trap information:

```
RALPROFS     PROFILE  IC=LOGONX
             AUTH    MSGRECVR=YES,CTL=GLOBAL   5
             OPCLASS 1,2
               END
```

Figure 159. NetView operator profile: SYSPLEX.DSIPRF(RALPROFS)

5 With AUTH MSGRECVR=YES definition, the NetView operator who uses this profile can receive unsolicited messages. This authorization is required for SNMP traps to be shown on the console of this operator.

A TRAPSON command can be issued by a CLIST. After it is issued, traps can start to arrive asynchronously. And then the TRAP messages in the range from SNM030 to SNMP039 are sent to the authorized NetView operator. Note that, however, traps can be received by only one operator. The messages also go to the NetView log file, so that you can always browse the log file to see trap data received. You can also assign trap messages to go to a specific operator using the NetView ASSIGN operator command.

In the following screens, we will show several examples of SNMP traps captured by the SNMP query engine. The trap information consists of multiline messages, in which the first line is always message SNM030I and the last line is always message SNM039I.

The following common trap types exist:

```
0  coldStart
1  warmStart
2  linkDown
3  linkUp
4  authenticationFailure
5  egpNeighborLoss
6  enterpriseSpecific
```

```
= RA28N  %
SNM030I SNMP request 1028 received following trap:
SNM031I Agent Address: 9.24.104.149      6
SNM032I Generic trap type: 0             7
SNM033I Specific trap type: 0
SNM034I Time stamp: 100
SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.3.13
SNM039I SNMP request 1028 End of trap data
```

Figure 160. SNMP trap message: coldStart trap

6 tells you from which IP host this trap originates. 7 tells you what kind of trap this is (coldStart trap in this example).

```
= RA28N  %
SNM030I SNMP request 1028 received following trap:
SNM031I Agent Address: 9.24.104.149
SNM032I Generic trap type: 2                          8
SNM033I Specific trap type: 0
SNM034I Time stamp: 4700
SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.3.13
SNM036I Variable name: 1.3.6.1.2.1.2.2.1.1.8          a
SNM037I Variable value type: 1
SNM038I Variable value: 8
SNM036I Variable name: 1.3.6.1.2.1.2.2.1.7.8          b
SNM037I Variable value type: 1
SNM038I Variable value: 1
SNM036I Variable name: 1.3.6.1.2.1.2.2.1.8.8          c
SNM037I Variable value type: 1
SNM038I Variable value: 1
SNM039I SNMP request 1028 End of trap data
```

Figure 161. SNMP trap message: linkDown trap

Figure 161 shows the linkDown trap (8). This trap consists of several MIB objects that identify the interface whose status has changed from up to down. The MIB objects carried in this trap are ifIndex (a), ifAdminStatus (b), and ifOperStatus (c).

```
= RA28N  %
SNM030I SNMP request 1028 received following trap:
SNM031I Agent Address: 9.24.104.149
SNM032I Generic trap type: 4                    d
SNM033I Specific trap type: 0
SNM034I Time stamp: 104300
SNM035I Enterprise Object ID: 1.3.6.1.4.1.2.3.13
SNM039I SNMP request 1028 End of trap data
```

Figure 162. authenticationFailure trap

Figure 162 shows you the messages you would see when an authenticationFailure trap (d) arrives. To enable an SNMP agent to send the authenticationFailure traps, you might have to change the value of the snmpEnableAuthenTraps MIB object.

The possible value for this object is either enabled(1) or disabled(2).

```
* RA28N    snmp get 9.24.104.149 publicv1 snmpEnableAuthenTraps.0
+ RA28N    SNM050I SNMP Request 1018 from KAKKY accepted, sent to Query Engine
= RA28N
SNM040I SNMP Request 1018 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.11.30.0
SNM043I Variable value type: 1
SNM044I Variable value: 2                                          f
SNM049I SNMP Request 1018 End of response
--------------------------------------------------------------------------------


???
netvasis snmp get 9.24.104.149 publicv1 snmpEnableAuthenTraps.0     e
```

Figure 163. Verify the snmpEnableAuthenTraps MIB setting

By issuing the SNMP GET command (e) to the snmpEnableAuthenTraps MIB object, you can verify the current setting of this MIB object. If sending authenticationFailure traps has been disabled (f), issue the SNMP SET command (g) to change the MIB value to enabled (h) so that the traps can be sent to the SNMP manager.

```
* RA28N    snmp set 9.24.104.149 publicv1 snmpEnableAuthenTraps.0 1
+ RA28N    SNM050I SNMP Request 1020 from KAKKY accepted, sent to Query Engine
= RA28N
SNM040I SNMP Request 1020 from KAKKY Returned the following response:
SNM042I Variable name: 1.3.6.1.2.1.11.30.0
SNM043I Variable value type: 1              h
SNM044I Variable value: 1
SNM049I SNMP Request 1020 End of response
--------------------------------------------------------------------------------

???
netvasis snmp set 9.24.104.149 publicv1 snmpEnableAuthenTraps.0 1       g
```

Figure 164. Change the snmpEnableAuthenTraps MIB setting

Appendix B. Quality of Service (QoS)

SecureWay Communications Server for OS/390 V2R8 IP introduces new enhancements of the elements of Quality of Service (QoS). These new improvements include:

- Function of policy agent to serve as a central point of control/access for all policies.
- Function of policy agent to monitor a local file and LDAP server, and dynamically update policies.
- Reservation Protocol (RSVP) agent has been integrated into CS for OS/390.
- RSVP agent to use Policy API to obtain RSVP policies and apply policy control to flows.
- Stack to provide performance monitoring metric, such as round trip time (RTT), throughput for TCP or packet/byte counts for UDP.
- Performance monitor subagent to retrieve performance monitoring data from the stack and allow the monitoring of various QoS characteristics.
- Subagent to provide traps for monitor entities that violate boundaries established using PolicyMonitorTable MIB
- `NETSTAT` command to display policy definitions and performance data.

B.1 A short introduction to Quality of Service

In this section, we provide an overview of traffic priority, or Quality of Service (QoS). For detailed information, refer to *TCP/IP Tutorial and Technical Overview*, GG24-3376.

B.1.1 General concepts

Bandwidth, in the Internet and intranets of today, is an important subject. The amount of data that has to be transmitted through the Internet increases exponentially. Applications such as RealAudio, RealVideo, Internet Phone software, and video conference systems require a lot more bandwidth than the applications that were used in the early years of the Internet. Common Internet applications, such as Telnet, FTP or World Wide Web, cannot tolerate packet loss but are less sensitive to variable delays. Most real-time applications, on the other hand, show just the opposite behavior, meaning they can compensate for a reasonable amount of packet loss but are usually very critical toward high variable delays.

This means that without any bandwidth control, the quality of these real-time streams depends on the only available bandwidth. Low bandwidth, or better unstable bandwidth, causes bad quality in real-time transmissions, for instance, dropouts and hangs. The quality of a transmission using real-time protocol RTP depends on the utilization of the underlying IP delivery service.

A Quality of Service (Qos) is the overall service that a user or an application receives from a network, in terms of availability, throughput, delay, etc. The basic IP protocol stack provides only one QoS which is called *best-effort*. A packet is routed from point to point without any guarantee for a specific bandwidth or

minimum time delay. Based on the best-effort traffic model, Internet requests are handled a *first come, first serve* basis. This means that all requests have the same priority and are handled one after the other. To make bandwidth reservations for specific connections or to raise the priority for a special request is not possible. New strategies, therefore, were developed to provide predictable services for the Internet.

Today, there are two main technologies to provide QoS on the Internet. Figure 165 illustrates two types of QoS.

- Integrated Services

  End-to-end reservation-based service that uses RSVP to request an appropriate level of service for a specific session/flow. It provides explicit end-to-end QoS to an application along the data path from sender to receiver.

- Differentiated Services

  Differentiated Services is a type of service that provides implicit QoS for different classes of traffic or users. It provides quasi-static service differentiation between broad classes of users and applications. With Differentiated Services, you define a certain service level for a generic service type, such as all FTP traffic or all Telnet traffic between two network entities.

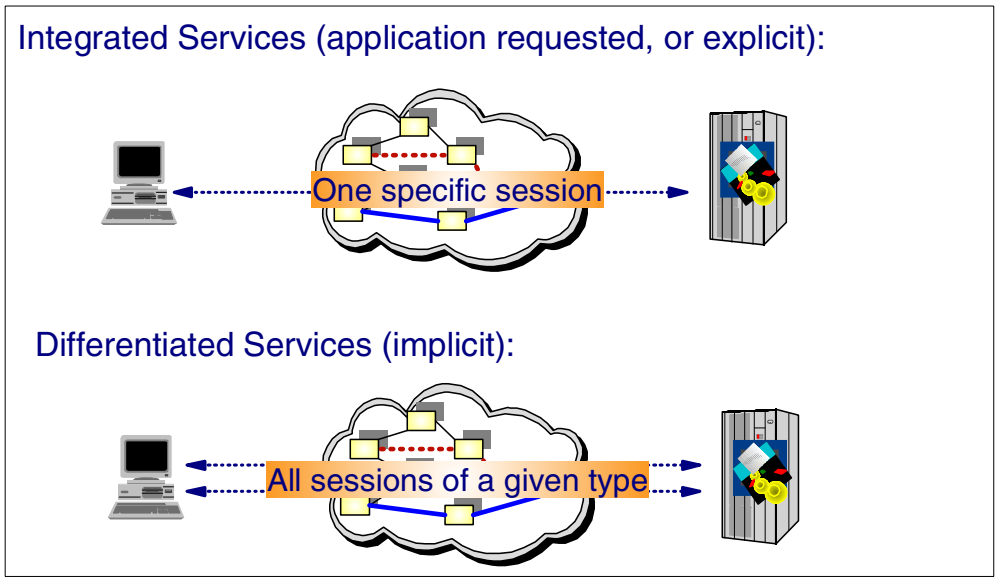


Figure 165. Type of QoS

B.1.2 Integrated Services

Integrated Services (IS) was developed to optimize network and resource utilization for new real-time applications, such as real-time multimedia, which requires QoS guarantees. IS controls traffic flows from sender to receiver all over the path. A flow can be defined as a distinguishable stream of related datagram from a unique sender to a unique receiver, of a single user activity, and requires the appropriate QoS.

For example, a flow might consist of one video stream between a given host pair. To establish the video connection in both directions, two flows are necessary. Each application that initiates data flows can specify which QoS is required for

this flow. If a videoconference tool needs a minimum bandwidth of 128 kbps and a minimum packet delay of 100 ms to ensure a continuous video display, such QoS can be reserved for this connection.

To support the Integrated Services model, an Internet router has to be able to provide an appropriate service as requested by each flow, in accordance with the service model. The Integrated Services model uses the Reservation Protocol to define and control QoS reservations.

Figure 166 shows the operation of the Integrated Services model into a host and a router:

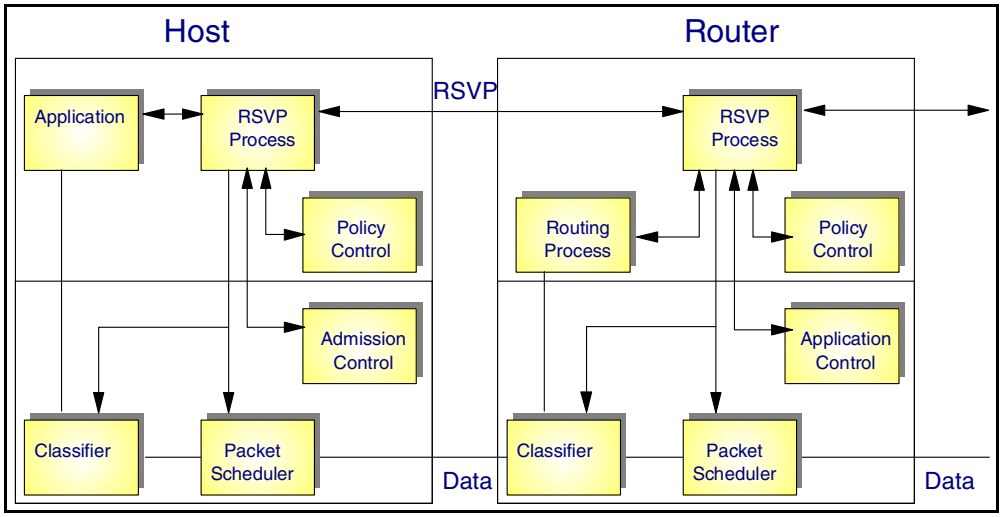


Figure 166. Integrated Services model

Packet scheduler

The packet scheduler manages the forwarding of different packet streams in hosts and routers, based on their service class, using queue management and various scheduling algorithms.

Packet classifier

The packet classifier identifies packets of an IP flow in hosts and routers that will receive a certain level of service.

Admission control

The admission control contains the decision algorithm that a router uses to determine if there are enough routing resources to accept the requested QoS for a new flow.

The Reservation Protocol (RSVP)

The Reservation Protocol (RSVP) is a model used by Integrated Services to set up and control QoS reservations. RSVP is defined in RFC 2205 and has the status of a proposed standard. Since RSVP is an Internet control protocol and not a routing protocol, it requires an existing routing protocol to operate, and must be implemented in all routers on the reservation path. The key concepts of RSVP are flows and reservation.

Service classes

The Integrated Services model uses different classes of service that are defined by the integrated services IETF working group. Depending

on the application, those service classes provide tighter or looser bounds on QoS controls. The current IS model includes the Controlled Load Service, which is defined in RFC 2211, and the Guaranteed Service, defined in RFC 2212.

Controlled Load Service

The Controlled Load Service is intended to support the class of applications that are highly sensitive to overloaded conditions in the Internet, such as real-time applications. These applications are working well on unloaded networks but degrade quickly under overloaded conditions. If an application uses the Controlled Load Service, the performance of a specific data flow does not degrade if the network load increases.

Guaranteed Service

The Guaranteed Service model provides functions that ensure that datagrams will arrive within a guaranteed delivery time. This means that every packet of a flow that conforms to the traffic specifications will arrive at least at the maximum delay time that is specified in the flow descriptor. Guaranteed Service is used for applications that need a guarantee that a datagram will arrive at the receiver not later than a certain time after it was transmitted by its source.

For more information on Integrated Services, refer to *TCP/IP Tutorial and Technical Overview*, GG24-3376.

B.1.3 Differentiated Services

Differentiated Services (DS) was developed to get a possibility to provide differentiated classes of service for Internet traffic, to support various types of applications, and specific business requirements. DS offers predictable performance, such as delay, throughput or packet loss, for a given load at a given time. Different from Integrated Services, DS doesn't use per-flow signalling. DS provides scalable service discrimination in the Internet without the need for per-flow state and signaling at every hop. It is not necessary to perform a unique QoS reservation for each flow. With DS, the Internet traffic is divided into different classes with different QoS requirements, and different service levels can be allocated to different groups of Internet users. Figure 167 on page 181 illustrates an analogy of traffic grouping.

Well-known TCP/IP applications can be grouped in the following categories:

High-Priority Traffic

- Telnet
- CICS transaction
- SAP R/3
- DB2 connect
- SNMP

Medium-Priority Traffic

- HTTP (Web)
- Mail
- SAP R/3
- Message Queuing

- ADSM Restores

Low-Priority Traffic

- FTP
- ADSM Backup
- REXEC/RSH

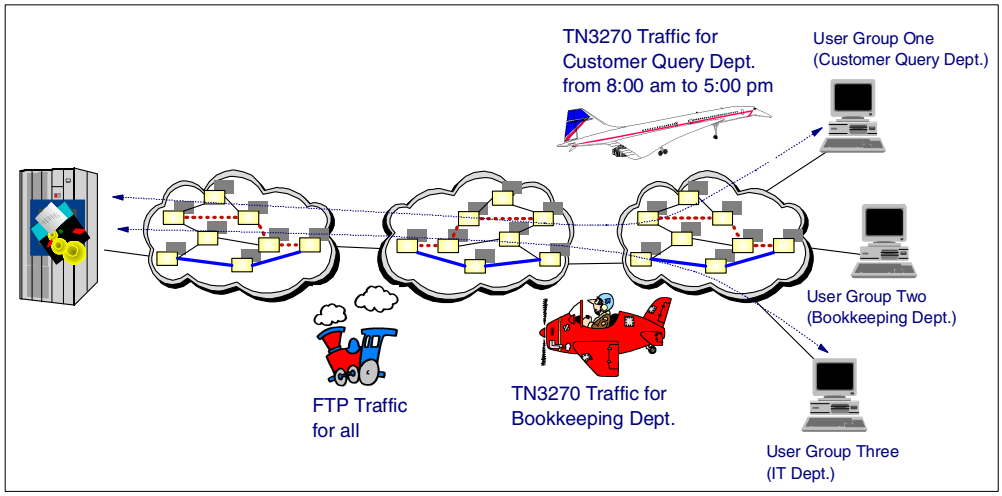


Figure 167. Grouping of traffic

B.1.3.1 Differentiated Services architecture

QoS guarantees with Differentiated Services are static and stay long term in routers. This means that applications using DS do not have to specify QoS reservations for a specific QoS. All traffic that passes DS-capable networks can receive a specific QoS. The data packets must be marked with the DS byte that is interpreted by the routers in the network.

DS byte

The DS byte is used to distinguish the data packets from different customers in DS-capable network. Each packet is modified in a specific field. A small bit pattern, called the DS byte, in each packet, is used to mark the packet that it receives a particular forwarding treatment at each network node.

The DS byte uses the space of type of service (TOS) octet in the IPv4 header, and the traffic class octet in the IPv6 header. All network traffic inside a domain receives a service depending on the traffic class that is specified in the DS byte.

To provide SLA conform services, the following mechanisms have to be combined in a network:

- Setting bits in the DS byte (TOS octet) at network edges and administrative boundaries.
- Using those bits to determine how packets are treated by the routers inside the network.
- Conditioning the marked packets at network boundaries in accordance with the QoS requirements of each service.

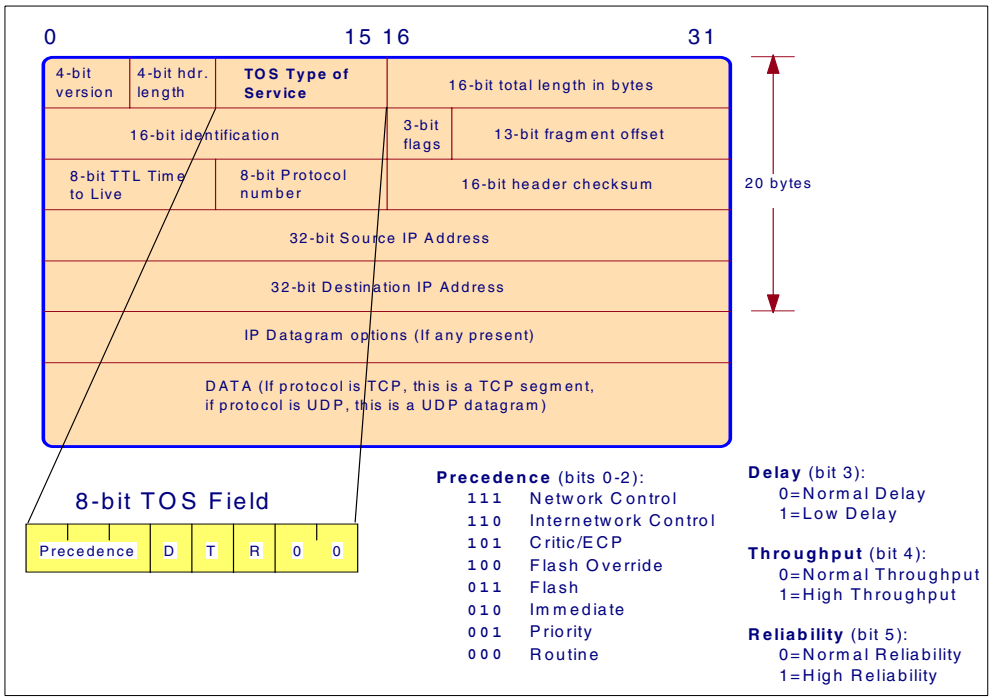
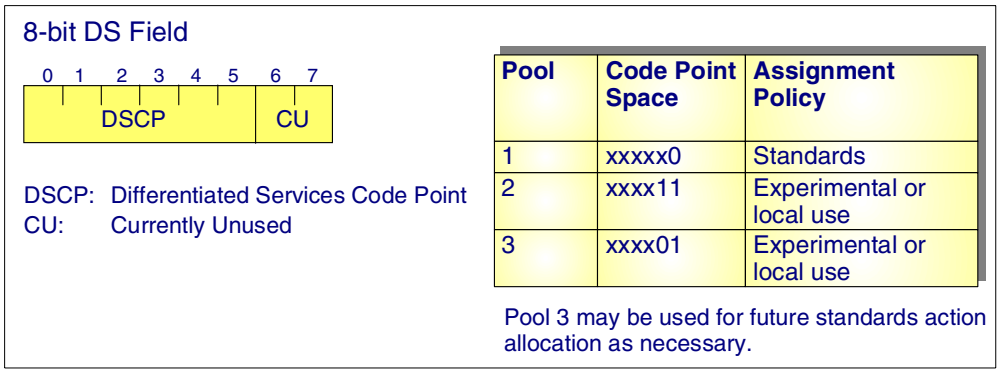


Figure 168. IPv4 datagram and TOS field (RFC 791 and RFC 1812)

Of the six bits of DS byte (TOS octet), the first three bits indicate the precedence: 0 for normal to 7 for network control.The next three bits show delay, throughput and reliability. If the delay bit is set, the packet should be processed with low delay. If the throughput bit is set, the packet should be processed with high throughput. And if the reliability bit is set, the packet should be processed with high reliability.



Pool	Code Point Space	Assignment Policy
1	xxxxx0	Standards
2	xxxx11	Experimental or local use
3	xxxx01	Experimental or local use

Figure 169. Differentiated services field (RFC2474)

Figure 169 shows the DS field redefined in RFC 2474. The six bits of the DS field are used as Differentiated Services Code Point (DSCP) to select the traffic class that a packet experiences at each node. The two-bit currently unused (CU) field is reserved and can be assigned in the future. The DS byte specifications supersede the TOS octet definitions of RFC 1349.

To allow filtering based on bits of the DS byte, the following are to be defined.

1. The filter mask - which of the eight bits are to be filtered. If only the precedence bits to be filtered, then set the mask to E0.
2. The value of the bits to be changed.

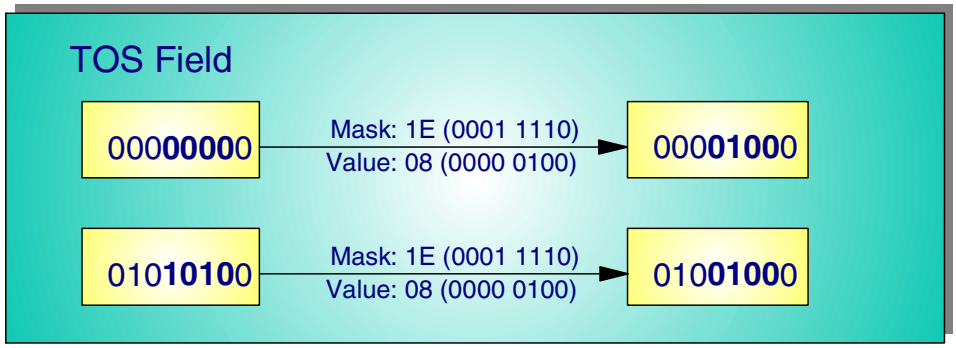


Figure 170. DS byte changing

Figure 170 shows an example of the mask and the value to be changed in the DS byte.

The per-hop behavior

Each DS-capable network device must have information on how packets with different DS bytes should be handled. In the DS specifications this information is called the per-hop behavior (PHB). It is a description of the forwarding treatment a packet receives at a given network node. The DSCP value in the DS byte is used to select the PHB a packet experiences at each node. To provide predictable services, per-hop behaviors need to be available in all routers in a Differentiated Services-capable network. The PHB can be described as a set of parameters inside a router that can be used to control how packets are scheduled onto an output interface. This can be a number of separate queues with set priorities, parameters for queue lengths or drop algorithms, and drop preference weights for packets.

Figure 171 on page 184 shows how routers should use the precedence bits in the DS byte.

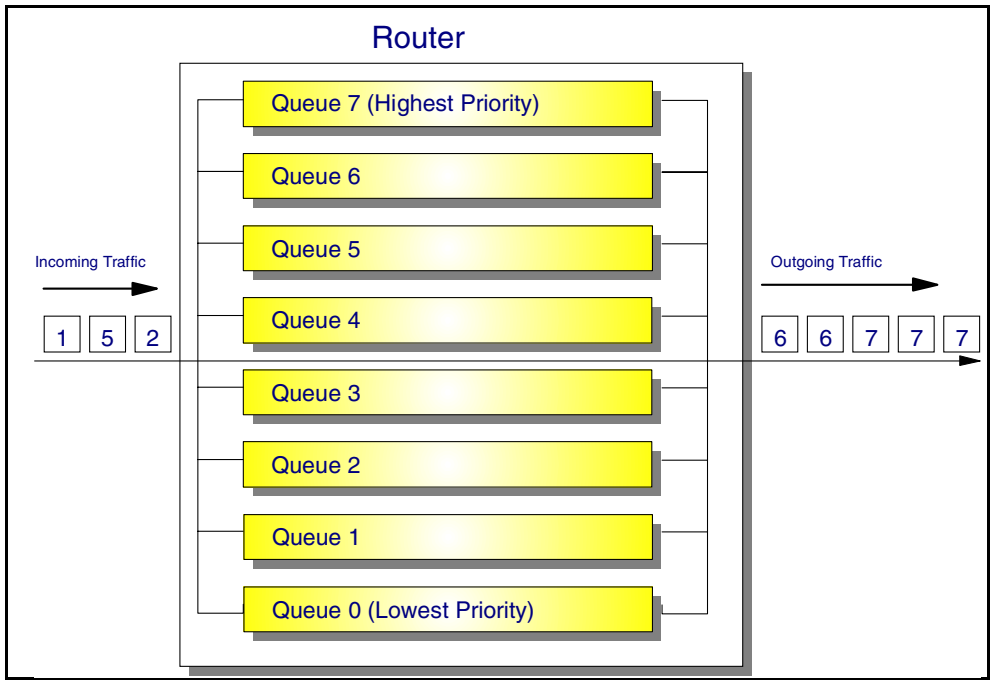


Figure 171. DS routing example

For more information on Differentiated Services, refer to *TCP/IP Tutorial and Technical Overview*, GG24-3376.

B.1.3.2 Service level agreement

Service level policy is a method for controlling and regulating service differentiation. Together, service differentiation and service level policy form an integral part of the service level agreement (SLA) function that has become increasingly important as TCP/IP networks evolve. SLA is a service contract between a customer and a service provider that specifies the details of the traffic classifying and the corresponding forwarding service a customer should receive. A customer may be a user organization or another DS domain. The service provider must ensure that the traffic of a customer, with whom it has an SLA, gets the contracted QoS. Therefore, the service provider's network administration must set up the appropriate service policies and measure the network performance to guarantee the agreed traffic performance.

B.1.3.3 Lightweight Directory Access Protocol

The administrative policies can be stored in a directory server function in a standardized format. With a standardized mechanism, this directory information can be provided to the Differentiated Services boundary components. These components act as clients for these policies distribution functions. These functions are provided by LDAP.

In CS for OS/390 V2R8, now we can administer advanced policies or update existing policies dynamically using LDAP commands facility without restarting the policy agent.

Figure 172 on page 185 illustrates how a Quality of Service is built in a network using SLA, LDAP, and the Service Policy.

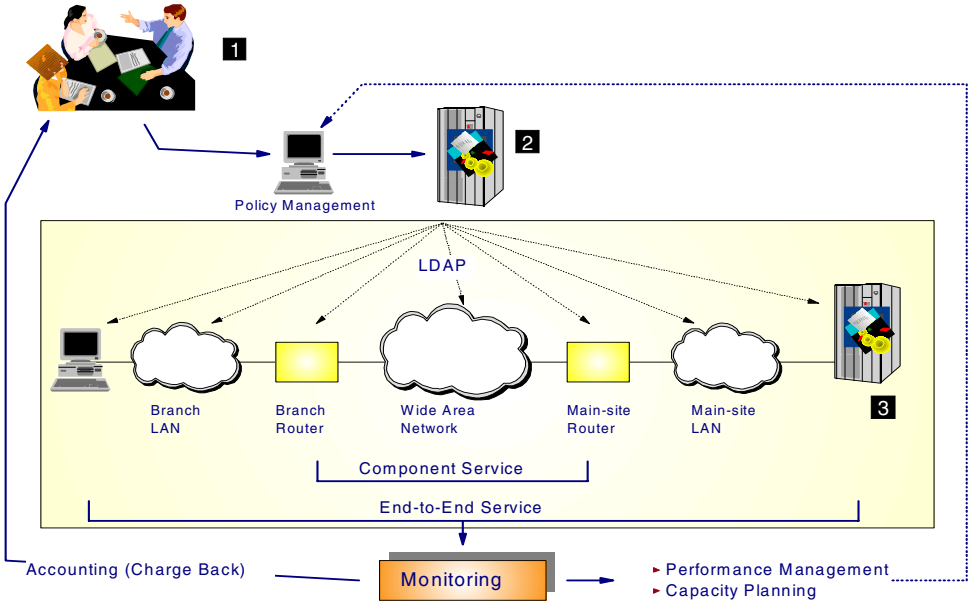


Figure 172. The big service policy management

1 Business units and IT define in business the required service = service level agreement (SLA) - applications, users, transaction volumes, response times, and availability

2 Service policies are encoded and stored in an LDAP directory or flat file

3 Service policy enforcement and monitoring in all network components

For more information on Lightweight Directory Access Protocol, refer to *Understanding LDAP*, SG24-4986.

B.1.4 QoS implementation in CS for OS/390

CS for OS/390, to provide Quality of Service, has made an evolutionary solution from V2R6 to V2R8. CS for OS/390 has improved not only the Differentiated Services delivery but also the Integrated Services delivery.

In delivering Integrated Services, CS OS/390 V2R6 made RSVP agent available through the Web to provide RSVP functionality. In addition, RSVP agent provided RSVP API which uses AF_UNIX socket communication between a QoS-aware application and RSVP agent. For CS OS/390 V2R7, the enhancements included Request/Delete reservations over a particular interface (initially support for ATM only) and a stack's capability to return an application's route and to notify RSVP agent when routes or interfaces were added/deleted.

In delivering Differentiated Services, CS for OS/390 V2R6 supported the policy agent available through the Web to be able to install policies from a local file into a stack policy table. Policy statements can also cause the TOS byte to be set for outgoing packet. The TOS byte is used by network routers to provide varying levels of service, and it is also used by Queued Direct I/O interfaces on OS/390 to map priority levels. Another enhancement in Release 6 was that the policy is selected in the IP layer, so every IP packet is examined for policy applicability.

The policy agent, in CS OS/390 V2R7, was integrated into CS OS/390 and could install the policies from Lightweight Directory Access Protocol (LDAP) server and a local file. As in Release 6 the policy filtering was done in the IP layer; in Release 7 the policy was selected in the TCP/UDP/RAW layers to improve performance. Another QoS improvement in Release 7 was that TCP data rates and connection limits were enforced through the control of the congestion window and the receive buffer size.

Many more enhancements for delivering IS and DS are made to implement QoS in CS OS/390 V2R8. In Release 8, RSVP agent is integrated into CS for OS/390; this means that RSVP is a component of CS OS/390. Furthermore, RSVP agent interfaces with policy agent using a new Policy API to obtain RSVP policies and apply policy control to flows.

The policy agent, in Release 8, can negotiate RSVP information with RSVP agent through a new policy agent API. This means that now the policy agent serves as a central point of control/access for all policies. In addition, it monitors changes to policy definitions and updates the stack policy table dynamically.

In addition to enhancements associated with RSVP agent and the policy agent, CS for OS/390 V2R8 also provides enhancements for performance monitoring. The TCP/IP stack provides performance monitoring metrics, such as RTT, throughput, etc. for TCP, and packet/byte count for UDP. And for monitoring performance, a new SNMP subagent, the Service Level Agreement(SLA) SNMP subagent, has been introduced. This subagent retrieves performance monitoring data from the stack, and allows the monitoring of various QoS characteristics. Also, the subagent provides traps for monitored entities, for example policies or applications, that violate boundaries that are established via PolicyMonitorTable Management Information Base (MIB). Another one related to monitoring performance enhancement is that the NETSTAT command can be used to display policy definitions and performance data.

Figure 173 on page 187 shows the QoS component in CS for OS/390 V2R8 IP.

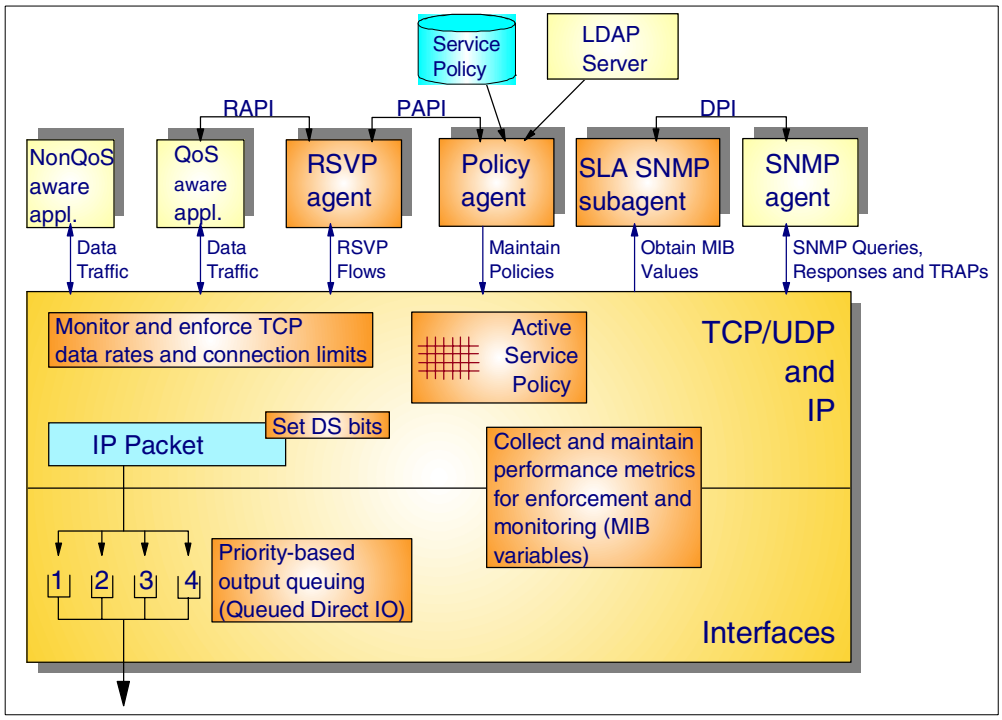


Figure 173. QoS components in CS for OS/390 V2R8 IP

B.2 OS/390 UNIX service policy agent

The OS/390 UNIX service policy agent (PAGENT) is a server responsible for reading the service level policies from a configuration file and installing them in the TCP/IP stack.

Network administrators can use the OS/390 UNIX service policy agent (PAGENT) to define service level policies for their users' benefits of the service level policies:

- Control of TCP throughput.
- Admission control for connection requests from clients.
- Blocking of unwanted datagrams.
- Activation of specification of policies depending on date and time.
- Prioritization of all IP traffic both within CS for OS/390 and IP routers that use TOS settings for priority queuing and selective discards.
- From OS/390 V2R8 IP, policy agent provides limits for RSVP reservation parameters requested by RSVP applications, and limiting total active RSVP traffic flows.

In Figure 173 it can be seen that the policy agent retrieves service policy rules and statements from a policy configuration file or from the LDAP server and installs them in the CS for OS/390 stack (across all MVS images).

PAGENT retrieves service policy rules and statements from a policy configuration file or from a Lightweight Directory Access Protocol (LDAP) server and installs them in the CS for OS/390 stack. PAGENT sets the TOS value of outgoing IP packets, and it can be used to flush out existing policies in the stack or update

them as necessary. PAGENT also logs information and error messages into a HFS file. The policy agent also monitors the local file and LDAP server dynamically for updated policies maintained in the TCP/IP stack.

Note: Service Policy works only for the MVS as host, not for the routing part. Datagrams passing through are not controlled by the Service Policy.

Queued Direct Input/Output (QDIO) uses the TOS settings as the basis of its queuing structure (one queue for each priority -- Network, High, Medium, and Low) and is therefore dependent on a service level policy to generate the proper TOS settings for all types of IP traffic. The LINK IPAQGNET statement is used to define a gigabit Ethernet link on an MPCIPA device belonging to a QDIO interface. At the time of writing, this is the only interface supported by TCP/IP that uses QDIO.

B.2.1 Implement policy agent in CS for OS/390 V2R8 IP

At ITSO Raleigh, we configured policy agent in two ways. In one scenario, the policy agent extracts the policy information from a static configured HFS file, and in another case, an LDAP server running in OS/390 provides all policy information in the network.

B.2.1.1 Policy agent configuration file

The policy configuration file is used when PAGENT is started. This initial configuration file can be used to point to other policy files that contain specific policies for other corresponding TCP/IP images. In one scenario, or the first PAGENT implementation, we used a local configuration file resides in a HFS file. In the other one, the LDAP server is used to extract the policy information. These two configuration files will be shown later. See *OS/390 Security Server LDAP Server Administration and Usage Guide*, SC24-5861 for the information about the LDAP server at ITSO Raleigh.

The following statements may be configured in a policy agent configuration file:

- LogLevel statement: This statement specifies the level of logging or tracing.
- TcpImage statement: This statement specifies a TCP/IP stack name and its associated policy control file to be installed in that image.
- ReadFromDirectory statement: This statement initializes PAGENT as an LDAP client, so that the rules will be downloaded from the LDAP server in addition to being read from this configuration file.
- SetSubnetPrioTosmask statement: This statement defines the TOS/priority field in the IP header type of service byte. It is used by the TCP/IP stack to read the TOS value and assign appropriate service to the corresponding IP packets.
- ServiceCategories statement: This statement specifies the type of service that a flow of IP packets should receive end-to-end as they pass through the network.
- ServicepolicyRules statement: This statement specifies characteristics of IP packets that are used to match a corresponding service category, that is it defines a set of IP datagrams that should receive a particular service.

Although the /etc/pagent.conf is the default configuration file, a specific search order is used when starting the service policy agent. The following order is used:

- File or data set specified with the -c startup option
- File or data set specified with the PAGENT_CONFIG_FILE environment variable
- /etc/pagent.conf
- hlq.PAGENT.CONF

PAGENT configuration file with the policy definitions

In the first scenario, the policy definitions have been configured statically in the PAGENT configuration file as shown below:

```
#  SecureWay Communications Server IP for OS/390
#  5647-A01 (C) Copyright IBM Corp. 1998, 1999.
#  Licensed Materials - Property of IBM
#
#       /etc/pagent2.conf  (PAGENT policy configuration)
#
# LogLevel Statement
Loglevel 255              1

# TcpImage Statement
TcpImage TCPIPA /etc/pagent2.conf  FLUSH  2

# ReadFromDirectory  statement
#
# ReadFromDirectory       3
# {
#   LDAP_Server            172.16.250.3
#   LDAP_Port              389
#   Base                   o=IBM_US,c=US
#   LDAP_SelectedTag       MVS03A
# }


# SetSubnetPrioTosMask  statement   4
{
SubnetAddr         9.24.104.76   A
SubnetTosMask      11100000 3    B
PriorityTosMapping 4 00000000    C
PriorityTosMapping 4 00100000
PriorityTosMapping 3 01000000
PriorityTosMapping 2 01100000
PriorityTosMapping 1 10000000
PriorityTosMapping 1 10100000
PriorityTosMapping 1 11000000
PriorityTosMapping 1 11100000
}

# ServiceCategories statement
ServiceCategories   networkcontrol   5
{
     OutgoingTOS    11100000          # Precedence bits (first 3 bits)
}

ServiceCategories   interactive1
{
     OutgoingTOS    10000000
}
```

```
ServiceCategories   batch1
{
     OutgoingTOS    01000000
}

ServiceCategories   Tscsrv      D
{
      OutgoingTOS    11000000   E
      MaxRate          120
      MaxTokenBucketPerFlow 440
      Maxdelay       10
}

# ServicePolicyRules statement

ServicePolicyRules  ospf       5    # OSPF link advertisement traffic
{
     ProtocolNumber     89          # OSPF protocol number
     ServiceReference   networkcontrol
     PolicyName     ospf-mvs
}

ServicePolicyRules  ftpd           # FTP traffic
{
     ProtocolNumber     TCP
     Direction          Both
     SourcePortRange    20 21       # Both FTP control and data ports
     ServiceReference   batch1   F
     PolicyName         ftp-mvs
}

ServicePolicyRules  telnetd        # telnet traffic
{
     ProtocolNumber     TCP
     Direction          Both
     SourcePortRange    23
     ServiceReference   interactive1
     PolicyName         tn3270
}
ServicePolicyRules  TscsrvRule
{
     Direction           both
     PolicyScope         DataTraffic
     ProtocolNumber      TCP
     SourcePortRange     12000
     SourceAddressRange  172.16.232.1 172.16.252.254
     ServiceReference    Tscsrv
     PolicyName          TscsrvP
}
```

1 The log level is set with the integer that specified the level of logging/tracing. We are using the LogLevel 255, that means all messages except trace messages are captured. The supported levels are:

- 1 - SYSERR - System error messages
- 2 - OBJERR - Object error messages
- 4 - PROTERR - Protocol error messages
- 8 - WARNING - Warning messages

- 16 - EVENT - Event messages
- 32 - ACTION - Action messages
- 64 - INFO - Informational messages
- 128 - ACNTING - Accounting messages
- 256 - TRACE - Trace messages

2 In the TcpImage statement, up to four parameters can be configured. The first one specifies the TCP/IP stack name to which the policy agent establishes an affinity. The next one is the path of the policy control file. The policy definitions could be configured in a file other than the policy configuration file itself.

For the third parameter, you can specify if the policy agent deletes all the policies existing in the TCP/IP stack when it is started. Note that the policies installed in the TCP/IP stack will be deleted at PAGENT startup time only if the FLUSH parameter is specified. This prevents the policies from being deleted unexpectedly if PAGENT terminates abnormally. If you want to remove policies when you cancel PAGENT, you could restart PAGENT afterward, pointing to a configuration file with FLUSH specified but no policies defined.

In the last parameter, specify the time interval in seconds for checking the creation or modification time of the configuration file(s) and for refreshing policies from the LDAP server. The default value is 1800 seconds (30 minutes). Note that dynamic monitoring for the configuration file is only supported for HFS files. MVS data sets are not monitored for changes.

3 The ReadFromDirectory statement initializes the policy agent as an LDAP client. Therefore if you use LDAP servers to store the policy information, you must have this configured. For this statement, several parameters that identify the LDAP server may be configured.

4 Define the TOS byte in the IP header using this statement.

The current IPv4 TOS byte format defines the first three bits to the precedence bits (for example, priority). Our default for the subnet TOS mask, if this statement is not specified, is 11100000. In this release only QDIO devices in OS/390 can support priorities. This statement is issued to set up TOS-to-priority mapping for those. QDIO supports four priority levels, 1 through 4, with 4 being the lowest priority.

In our sample configuration, we have configured:

A The local subnet interface address. 0.0.0.0 means the mask is the same for all interfaces

B The TOS mask

C Define the mapping between the TOS value and the priority level. This keyword can be repeated for each priority.

The default mappings are:

```
TOS             Priority
00000000            4
00100000            4
01000000            3
01100000            2
10000000            1
10100000            1
```

```
11000000          1
11100000          1
```

5 In the ServiceCategories statement you can specify various parameters that identify the type of service for a flow, such as the name of the service category (D) and the TOS value (E). Note that policies are applied to TCP on a connection basis, whereas they are applied to UDP/RAW on a per-packet basis.

A lot of interesting parameters can be specified with this statement:

- The maximum rate allowed for traffic in this service class
- The minimum rate/throughput
- The local transmission priority
- The local IP subnet for which this service category applies
- Maximum end-to-end delay time
- The maximum number of end-to-end connections at any instance of time
- Days in a week that this service policy is active
- The time of each day during which this service policy is active
- The type of service being requested by RSVP applications: Controlled Load (the default) or Guaranteed.
- Parameters that limit the traffic rate for RSVP flows.
- Maximum number of reserved flows allowed for RSVP applications

In our test we have mainly used the service categories named: networkcontrol, interactive1, batch1, and Tscsrv.

6 In the ServicePolicyRules statement, you can define a group of IP datagrams that should receive a particular service. The TCP/IP applications specified in the ServicePolicyRules are under the control of the policy agent according to the categories specified in the ServiceReference (F).

You may use the following parameters to identify the traffic or the scope in which this rule is applied.

- PolicyScope: Indicates to what traffic this policy rule applies. Valid values are DataTraffic, RSVP, and Both.
- Direction: Specify the direction of traffic for which this policy rule applies using one of the values of Incoming, Outgoing, or Both.
- ProtocolNumber: Identify protocol running on top of IP with a one-byte field in the IP header.
- Interface: The local IP subnet for which this policy rule applies.
- SourceAddressRange: The local IP address range.
- DestinationAddressRange: The remote IP address range.
- SourcePortRange: The local port range.
- DestinationPortRange: The remote port range.
- DaysOfWeekMask: Specify the days in a week that this policy rule is active.
- TimeOfDayRange: Indicate the time of day during which this policy rule is active.

PAGENT configuration file as an LDAP client

In the second scenario, the policy agent takes the role of an LDAP client so that it will be able to obtain all the policy definitions from the LDAP servers. We have the following configuration file named /etc/pagent3.conf in the HFS file system:

```
#  SecureWay Communications Server IP for OS/390
#  5647-A01 (C) Copyright IBM Corp. 1998, 1999.
```

```
#  Licensed Materials - Property of IBM
#
#       /etc/pagent2.conf  (PAGENT policy configuration)
#

Loglevel 255
TcpImage TCPIPA /etc/pagent3.conf  FLUSH
ReadFromDirectory                          1
{
  LDAP_Server             9.24.104.113     2
  LDAP_Port               389              3
  Base                    o=IBM_US,c=US    4
# LDAP_SelectedTag        MVS39A           5
}
```

The ReadFromDirectory statement initializes the policy agent as an LDAP client. We configured only three statements. Using the ReadFromDirectory statement (1), we can specify the host name or IP address (2) and the port number on which the directory server is running (3), and the distinguished name of the subtree in the directory containing the policies (4).

Using the LDAP_SelectedTag parameter (5), you can specify a string used to select a subset of the policies under the base tree. By default, the first host name returned by the gethostname API is used as a selector tag.

You can choose if the LDAP session with the directory server should be kept open or closed during an update interval time by using the LDAP_SessionPersistent parameter. If the LDAP session update interval is small, specify *yes* for this parameter to reduce the overhead of opening the session for each query.

For more information on the parameters for the policy agent configuration, refer to *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513.

B.2.1.2 Consideration to define policy rules

When you define and code the Policy Rules Direction, Source, and Destination, you have to consider when policy rules are applied.

- The policies are applied at TCP connection setup.
- For UDP, a policy rule is applied every time a UDP datagram is being received or sent.
- For other protocols, such as ICMP, OSPF, etc., every time an IP datagram is being received or sent, the policy rules are applied.
- And the policies are remapped when the policy definitions are being updated or refreshed. The rules will be remapped for every ACK segment in a TCP flow to adjust for time-of-day related policies.

B.2.1.3 Policy agent startup

The policy agent can be started from the UNIX System Services shell or as a started task. At ITSO Raleigh, we used a started task procedure to start policy agent.

Note: Security product (for example, RACF) authority is required to start the policy agent. The following commands can be used to create the necessary profile and permit users to use it:

```
RDEFINE OPERCMDS (MVS.SERVMGR.PAGENT) UACC(NONE)
PERMIT MVS.SERVMGR.PAGENT ACCESS(CONTROL) CLASS(OPERCMDS) ID(userid)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

The following is a policy agent started task procedure that we used in our system:

```
//PAGENT   PROC
//*
//* SecureWay Communications Server IP
//* SMP/E distribution name: EZAPAGSP
//*
//* 5647-A01 (C) Copyright IBM Corp. 1999.
//* Licensed Materials - Property of IBM
//*
//PAGENT   EXEC PGM=PAGENT,REGION=0K,TIME=NOLIMIT,
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c /
//*            etc/pagent2.conf -d'
//       PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c /
//            etc/pagent3.conf -d'              1
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-d'
//*
//* Example of passing parameters to the program (parameters must
//* extend to column 71 and be continued in column 16):
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c /
//*            etc/pagent3.conf'
//*
//STDENV   DD DSN=TCPIP.TCPPARMS.R2612(PAG&SYSCLONE.ENV),DISP=SHR 2
//*STDENV   DD PATH='/etc/pagent2.env',PATHOPTS=(ORDONLY)
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

You can use environment variables, either configured in an MVS data set or HFS file, specified by the STDENV DD (1) to run with the desired configuration. We have configured environment variables in an MVS data set (2), `TCPIP.TCPPARMS.R2612(PAG39ENV)` which contains:

```
 BROWSE    TCPIP.TCPPARMS.R2612(PAG39ENV) - 01.15
 Command ===>
******************************** Top of Data ********
LIBPATH=/lib:/usr/lib:/usr/lpp/ldapclient/lib:.   3
 PAGENT_LOG_FILE=/tmp/pagent.log                  4
TZ=EST5EDT                                        5
******************************** Bottom of Data ******
```

Figure 174. Environment variables for policy agent

We have configured three environment variables for the policy agent to run successfully. The first variable, LIBPATH, enables PAGENT to search the dynamic link libraries needed to act as an LDAP client (3). The PAGENT_LOG_FILE specifies the log file name used by PAGENT (4). Note that in our environment we have to put a single space at the head of the line, otherwise PAGENT creates an HFS file with an invalid name and we cannot browse/edit it.

The third one is significant if you want to control the policy using the time of day. For the policy agent to run in your local time zone, you might have to specify the time zone in your working location using the TZ environment variable (3) even if

you have the TZ environment variable configured in /etc/profile. Note that most OS/390 UNIX applications that start as MVS started tasks cannot use environment variables that have been configured in /etc/profile.

Note that, while we do not have the RESOLVER_CONFIG variable configured, PAGENT can establish an affinity to proper TCP/IP stack. The policy agent will use the TCP/IP image name configured in the TcpImage statement in the policy agent configuration file to determine to which TCP/IP it shall install the policies.

Special consideration on the policy agent running as an LDAP client
The policy agent running as an LDAP client has to use the dynamic link libraries, named ldap.dll, to access the LDAP servers. However, OS/390 V2R7 of the LDAP server was the last release in which ldap.dll was available.

Therefore for PAGENT to run in OS/390 V2R8, you have to create the ldap.dll library as a symbolic link to the new libraries.

To do this, complete the following sequence:

```
IRIANA @ RA39:/>ln -s /usr/lpp/ldapclient/lib/GLDCLDAP /usr/lib/ldap.dll  1
IRIANA @ RA39:/>ln -s /usr/lpp/ldapclient/lib/GLDCLDAP.x /usr/lib/ldap.x  1
IRIANA @ RA39:/>ls -laF ldap*                                             2
lrwxrwxrwx 1 OMVSKERN SYS1   28 Jun 29 14:23 ldap.dll@ -> /usr/lpp/ldapclient/GLDCLDAP
lrwxrwxrwx 1 OMVSKERN SYS1   30 Jun 29 14:24 ldap.x@ -> /usr/lpp/ldapclient/GLDCLDAP.x

IRIANA @ RA03:/>cd /usr/lpp/ldapclient/lib
IRIANA @ RA03:/usr/lpp/ldapclient/lib>extattr +a GLDCLDAP GLDCMMN     3
IRIANA @ RA03:/usr/lpp/ldapclient/lib>ls -E                               4
total 1560
-rwxr-xr-x  a-s  2 OMVSKERN SYS1      360448 Apr 20 17:14 GLDCLDAP
-rw-r--r--  --s  2 OMVSKERN SYS1        7200 Apr 20 17:14 GLDCLDAP.x
-rwxr-xr-x  a-s  2 OMVSKERN SYS1      430080 May 11 15:40 GLDCMMN
```

1 Create symbolic links for ldap.dll and ldpa.x. Using the `ls -laF` command, you can see the original files of the symbolic links created (2).

The original LDAP libraries have to have the APF authorized attribute to be loaded by PAGENT successfully. Therefore, you might have to change the attributes using the `extattr` command (3). The `ls -E` command shows you the extended attributes for HFS files. The `a` in the second column indicates that this file has APF authorized attributes.

If the dynamic link libraries do not have the APF authorized attribute, PAGENT fails and the following message is written in the SYSOUT DD.

```
CEE3512S An HFS load of module ldap.dll failed. The system return code was 0000000157;
 the reason code was 0BDF03AC.
 From entry point ReadLdapRules at compile unit offset +00000152 at address 14F20A1A.
```

Note, however, that to use the `extattr` command, the user has to have access to a specific RACF Facility class profile. Otherwise the command fails and you will see the following RACF message in your MVS console:

```
ICH408I USER(IRIANA  ) GROUP(OMVSGRP ) NAME(IRIANA RANTA        )
  BPX.FILEATTR.APF CL(FACILITY)
  INSUFFICIENT ACCESS AUTHORITY
  ACCESS INTENT(READ   )  ACCESS ALLOWED(NONE   )
```

This RACF message tells you that the user IRIANA requested the READ access to the RACF Facility class profile BPX.FILEATTR.APF. Therefore, issue the following commands and give the user access permission to the RACF profile:

```
PERMIT BPX.FILEATTR.APF CLASS(FACILITY) ID(userID) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

B.2.1.4 Policy agent log file

When you start the policy agent as a started task, the output message written to stdout and stderr goes to the data set or file specified with SYSPRINT or SYSOUT, respectively. But normally, PAGENT does not write output to stdout or stderr. Instead, output is written to the log file, which can be specified by the PAGENT_LOG_FILE environment variable and defaults to /tmp/pagent.log. When the -d parameter is specified, however, output is also written to stdout. The log file is created when the policy agent is activated, if it does not exist.

You can check whether your policies have been installed in an associated stack as defined. The following figures shows you the message after the policies have been installed. You have to specify option -d at startup, so additional messages will be displayed.

```
INFO   :.main[07]: Using log level 255    1
INFO   :....mailslot_create[07]: creating mailslot for timer
INFO   :...mailbox_register[07]: mailbox allocated for timer
TRACE  :..timer_initialize[07]:              TimerQ =1536e968
INFO   :..profile_initialize[07]: Pagent threads created installation(/flush) and
                                             initialization.
INFO   :...mailslot_create[07]: creating mailslot for terminate
INFO   :..pinit_init_tcpimages[06]: Creating temporary working file
                                            /tmp/TCPIPA.Pagent.tmp
INFO   :..pinit_init_tcpimages[06]: Creating temporary working file
                                            /tmp/TCPIPA.Pagent.tmp
INFO   :..pinit_init_tcpimages[06]: processed TcpImage statement:
                                       TcpImage TCPIPA /etc/pagent3.conf  FLUSH  2
INFO   :.check_main_config_file[06]: Finish starting policy profile installation(/flush) and
                                             initialization.
INFO   :.check_main_config_file[06]: Finish starting policy profile installation(/flush) and
                                             initialization.
INFO   :..mailbox_register[07]: mailbox allocated for dump
INFO   :...settcpimage[05]: Associate with TCP/IP image name = TCPIPA   3
INFO   :..FlushAllPolicies[05]: Start first by flushing all policies    4
INFO   :....settcpimage[05]: Associate with TCP/IP image name = TCPIPA  3
INFO   :...profile_delete_ALL_ServiceClass[05]: Image name: TCPIPA
INFO   :...profile_delete_ALL_ServiceClass[05]: Finished deleting
                                             ALL Service Class in image 0  5
INFO   :....settcpimage[05]: Associate with TCP/IP image name = TCPIPA
INFO   :...profile_delete_ALL_PolicyRule[05]: Image name:  TCPIPA
INFO   :...profile_delete_ALL_PolicyRule[05]: Finished deleting
                                             ALL Policy Rules    6
INFO   :..pinit_fetch_policy_profile[05]: Processing policy config data file:
                                          /etc/pagent3.conf for image TCPIPA   7
INFO   :..pinit_fetch_policy_profile[05]: Finish processing above policy config file  8
```

Figure 175. PAGENT startup log

During the initialization phase, PAGENT reads the policy agent configuration file and extracts the necessary information. In the sample above, the HFS file /etc/pagent3.conf has been specified in the -c start-up option for PAGENT.

The log level has been set to 255 (1), so that you will see additional information.

2 Processing the TcpImage statement. Since we have the FLUSH parameter configured, PAGENT will try to delete all existing policies (4,5) and rules (6) from the associated TCP/IP stack, that is TCPIPA (3) in this sample. After deleting all policies and rules (6), PAGENT will start to process the policy information that will be installed in the TCP/IP stack (7,8).

```
INFO   :..sla_ldap_get_service_rules[05]: Searching for rules with filter
        (&(objectclass=servicecategories)(TCPImageName=TCPIPA)(selectortag=MVS39A)) 1
INFO   :...enter_into_service_category_tab[05]: Entering attribute objectclass
INFO   :....parse_service_category_atts[05]: Unknown attribute objectclass in
        ServiceCategories - Ignoring.
INFO   :...enter_into_service_category_tab[05]: Entering attribute selectortag
INFO   :....parse_service_category_atts[05]: Unknown attribute selectortag in
        ServiceCategories - Ignoring.
INFO   :...enter_into_service_category_tab[05]: Entering attribute servicename
INFO   :...enter_into_service_category_tab[05]: Entering attribute tcpimagename
INFO   :....parse_service_category_atts[05]: Unknown attribute tcpimagename in
        ServiceCategories - Ignoring.
INFO   :...enter_into_service_category_tab[05]: Entering attribute policyscope
INFO   :...enter_into_service_category_tab[05]: Entering attribute maxrate
INFO   :...enter_into_service_category_tab[05]: Entering attribute minrate
INFO   :...enter_into_service_category_tab[05]: Entering attribute outgoingtos
          :
TRACE  :.ReadLdapRules[05]: Active time for SC=srv39a2, is = 441
TRACE  :..timer_set[05]:          1st queued=1538f4e0
INFO   :...settcpimage[05]: Associate with TCP/IP image name = TCPIPA 3
INFO   :..profile_install_A_ServiceClass[05]: Service Class: srv39a2
INFO   :..profile_install_A_ServiceClass[05]: Finished installing Service Class:srv39a2 2
TRACE  :.ReadLdapRules[05]: Active time for SC=srv39a1, is = 441
TRACE  :..timer_set[05]: put timer on existing Q, Qhead=1538f4e0
TRACE  :..timer_set[05]:          last queued=15390d88
INFO   :...settcpimage[05]: Associate with TCP/IP image name = TCPIPA 3
INFO   :..profile_install_A_ServiceClass[05]: Service Class: srv39a1
INFO   :..profile_install_A_ServiceClass[05]: Finished installing Service Class:srv39a1 2
```

Figure 176. Service classes installed in TCPIPA

Figure 176 1 shows that process of initialization of the policy agent. PAGENT is searching for all service categories stored in LDAP server (or local configuration file depending on the configuration), but it only looks for the service category objects associated with TCPIPA and MVS39A which is the hostname of TCPIPA on RA39. Note that PAGENT has gotten the TCP/IP stack name from its configuration, /etc/pagent3.conf, and the host name via the gethostname API.

PAGENT reads all entries of service categories, then installed them into TCPIPA. The policy agent finished installing service classes srv39a2 and srv39a1 2 to the TCPIP stack TCPIPA 3.

```
INFO   :..sla_ldap_get_policy_rules[05]: Searching for rules with filter
       (&(objectclass=servicepolicyrules)(TCPImageName=TCPIPA)(selectortag=MVS39A)) 4
INFO   :...enter_into_service_policy_table[05]: Entering attribute objectclass
INFO   :....parse_service_policy_atts[05]: Unknown attribute objectclass in
       ServicePolicyRules - Ignoring.
INFO   :...enter_into_service_policy_table[05]: Entering attribute selectortag
INFO   :....parse_service_policy_atts[05]: Unknown attribute selectortag in
       ServicePolicyRules - Ignoring.
INFO   :...enter_into_service_policy_table[05]: Entering attribute policyrulesname
INFO   :...enter_into_service_policy_table[05]: Entering attribute tcpimagename
INFO   :....parse_service_policy_atts[05]: Unknown attribute tcpimagename in
       ServicePolicyRules - Ignoring.
INFO   :...enter_into_service_policy_table[05]: Entering attribute policyscope
INFO   :...enter_into_service_policy_table[05]: Entering attribute daysofweekmask
INFO   :...enter_into_service_policy_table[05]: Entering attribute timeofdayrange
TRACE  :....parse_service_policy_atts[05]: Processing time intervals in: 7:00-20:00 5
TRACE  :....parse_service_policy_atts[05]:    From time: (7:00), (int=7:0)
TRACE  :....parse_service_policy_atts[05]:    To time: 20:00, (int=20:0)
TRACE  :....parse_service_policy_atts[05]: Finish processing time intervals in: 7:00-20:00
TRACE  :....parse_service_policy_atts[05]:    Active from: =7:0
TRACE  :....parse_service_policy_atts[05]:           To:   =20:0
INFO   :...enter_into_service_policy_table[05]: Entering attribute direction
INFO   :...enter_into_service_policy_table[05]: Entering attribute permission
INFO   :...enter_into_service_policy_table[05]: Entering attribute protocolnumber
INFO   :...enter_into_service_policy_table[05]: Entering attribute destinationportrange
INFO   :...enter_into_service_policy_table[05]: Entering attribute servicereference
INFO   :...enter_into_service_policy_table[05]: Entering attribute
            :
            :
INFO   :...UpdatePRQueue[05]: Weight computed for PR rule39a2 is: 2
INFO   :...UpdatePRQueue[05]: Weight computed for PR rule39a1 is: 2
INFO   :..UpdatePRProfileData[05]: caller id to this routine is 2
INFO   :..UpdatePRProfileData[05]: Weight computed for PR rule39a2 is: 2
INFO   :..UpdatePRProfileData[05]: Weight computed for PR rule39a1 is: 2
TRACE  :.ReadLdapRules[05]: Active time for PR=rule39a2, is = 201
TRACE  :..timer_set[05]: put timer on existing Q, Qhead=1538f4e0
TRACE  :..timer_set[05]:          last queued=153a424c
INFO   :...settcpimage[05]: Associate with TCP/IP image name = TCPIPA 6
INFO   :..profile_install_A_PolicyRule[05]: Policy Rule-rule39a2 src-0.0.0.0:0
       dst-0.0.0.0:20 proto-6 ServiceClass-srv39a2
INFO   :..profile_install_A_PolicyRule[05]: Finished installing policy rule: rule39a2 7
TRACE  :.ReadLdapRules[05]: Active time for PR=rule39a1, is = 21
TRACE  :..timer_set[05]: put timer on existing Q, Qhead=1538f4e0
TRACE  :..timer_set[05]:          new Qhead=153a4494
INFO   :...settcpimage[05]: Associate with TCP/IP image name = TCPIPA 6
INFO   :..profile_install_A_PolicyRule[05]: Policy Rule-rule39a1 src-0.0.0.0:12000
       dst-0.0.0.0:0 proto-6 ServiceClass-srv39a1
INFO   :..profile_install_A_PolicyRule[05]: Finished installing policy rule: rule39a1 8
INFO   :..sla_ldap_get_prio_rules[05]: Searching for rules with filter
    (&(objectclass=setsubnetpriotosmask)(TCPImageN ame=TCPIPA)(selectortag=MVS39A))
INFO   :...enter_into_subnet_tos_mask[05]: Entering attribute objectclass in index 0
         :
         :
```

Figure 177. Policy rules installed in TCPIPA

Figure 177 4 shows that PAGENT is processing service policy rules. It searches for all service policy rules stored in the LDAP server (or local configuration file), but it only looks for service policy rules associated with TCPIPA in MVS39A. PAGENT processes the interval time. 5 PAGENT sets up at what time the policy rules have to be applied. When PAGENT reads through all the service policy rules

entries, then installs them into TCPIPA. PAGENT installed policy rules rule39a2 7 and rule39a1 8 into the stack TCPIPA 6.

```
TRACE  :.....PolicyRuleTimer_expire[07]: PolicyRule rule39a1 (153a4278)
       current state (0 or 1) 1 9
TRACE  :.....PolicyRuleTimer_expire[07]: PolicyRule rule39a1 changes from
       Active state 1, to state 0 10
INFO   :.......settcpimage[07]: Associate with TCP/IP image name = TCPIPA
INFO   :......profile_delete_A_PolicyRule[07]: Policy Rule to be deleted:  rule39a1
INFO   :......profile_delete_A_PolicyRule[07]: Policy Rule was deleted by name
INFO   :......profile_delete_A_PolicyRule[07]: Finished deleting policy rule: rule39a1 11
TRACE  :......timer_set[07]: put timer on existing Q, Qhead=1538f4e0
TRACE  :......timer_set[07]:          last queued=153a4494
```

Figure 178. Timer expired and deleting rule

Figure 178 shows that after the policy rule timer expired, PAGENT changed the status of the policy rule rule39a1 from the current state which is active (1) 9 into not active (0) 10. Then the policy rule is deleted from TCPIPA 11.

B.2.1.5 Policy agent PID

The process ID (PID) of the policy agent can be found in the file /tmp/pagent.pid and looks like the following:

```
 BROWSE -- /tmp/pagent.pid ----
 Command ===>
******************************
838860825
******************************
```

B.2.1.6 Display policy definitions

In CS for OS/390 V2R8 IP, the NETSTAT command can be used to display the policy information. You can use the TSO NETSTAT or netstat command from the OS/390 UNIX shell. When using the TSO NETSTAT command you probably have to allocate SYSTCPD if it is not specified in your TSO logon procedures. Using NETSTAT SLAP REPORT TSO command, you can direct the output into a data set. By default it goes to *Userid*.NETSTAT.SLAP. Note, however, that the Display MVS console command does not support the NETSTAT SLAP option.

See *OS/390 SecureWay Communications Server IP User's Guide*, GC31-8514, for more information about these commands.

```
EZZ2350I MVS TCP/IP NETSTAT CS V2R8       TCPIP NAME: TCPIPA    1   16:41:39
EZZ2840I Policy:    RSVPappl      2
EZZ2841I   Profile:   rule39a3    3
EZZ2842I     Scope:             RSVP     4          Direction:        Both       5
EZZ2843I     Permission:        Allowed  6          Protocol:         0
EZZ2844I     LocalInterface:    0.0.0.0             Position:         0000000000
EZZ2845I     SourceIpFrom:      0.0.0.0             SourceIpTo:       0.0.0.0
EZZ2846I     SourcePortFrom:    000000              SourcePortTo:     000000
EZZ2847I     DestIpFrom:        0.0.0.0             DestIpTo:         0.0.0.0
EZZ2848I     DestPortFrom:      008000   7           DestPortTo:       008001     7
EZZ2849I     ServiceClass:      srv39a3  8
EZZ2850I       Scope:           RSVP     9           OutgoingTOS:      C0         10
EZZ2851I       Priority:        0                    Interface:        0.0.0.0
EZZ2852I       MaxRate:         0000000500  11       MinRate:          0000000300 12
EZZ2853I       MaxDelay:        0000000000           MaxConn:          0000000000
EZZ2854I     Performance Information:
EZZ2855I       FirstActTime:    15:07:04             LastMapTime:      00:00:00
EZZ2856I       TotalBytesIn:    0000000000           TotalBytesOut:    0000000000
EZZ2857I       BytesInDiscard:  0000000000           BytesOutDiscard:  0000000000
EZZ2858I       TotalInPackets:  0000000000           TotalOutPackets:  0000000000
EZZ2859I       ActConnMap:      0000000000           MaxConnLimit:     0000000000
EZZ2860I       AcceptConn:      0000000000           DeniedConn:       0000000000
```

Figure 179. Report from NETSTAT SLAP command

Figure 179 shows the report from the `NETSTAT SLAP` command and Figure 180 is the definition of the corresponding policy definition stored in the LDAP server. You will see how policy objects are mapped to the policy information installed in a TCP/IP stack:

```
r=rule39a3, o=IBM_US, c=US
objectclass=ServicePolicyRules
selectortag=MVS39A
policyrulesname=rule39a3                    3
policyname=RSVPappl                         2
tcpimagename=TCPIPA                         1
policyscope=RSVP                            4
daysofweekmask=0111110
timeofdayrange=7:00-20:00
direction=Both                              5
permission=Allowed                          6
destinationportrange=8000-8001              7
servicereference=srv39a3                    8
description=RuleDef for RSVP APPL on SYS39

s=srv39a3, o=IBM_US, c=US
objectclass=ServiceCategories
selectortag=MVS39A
servicename=srv39a3                         8
tcpimagename=TCPIPA                         1
policyscope=RSVP                            9
maxrate=500                                 11
minrate=300                                 12
outgoingtos=11000000                        10
flowservicetype=ControlledLoad
maxrateperflow=40
maxtokenbucketperflow=40
```

Figure 180. Corresponding policy definition in the LDAP server

B.2.1.7 Working with the policy agent

To illustrate the possibilities of the policy agent, we tested FTPD sessions with different TOS bytes and parameters. In one scenario, we used the PAGENT configuration file with the policy information in an HFS file. In another scenario, the LDAP server was used to store the policies.

Scenario 1

In the first test, we configured the policy definitions in the policy agent configuration file for FTP traffic as shown below:

```
ServicePolicyRules  ftpd            # FTP traffic
{
     ProtocolNumber     TCP
     Direction          Both
     SourcePortRange    20 21       # Both FTP control and data ports
     ServiceReference   batch1       1
}

ServiceCategories   batch1
{
OutgoingTOS    01000000            2
}
```

Figure 181. Policy definition for FTP traffic

In the ServiceCategories we specified the entry batch1 (1) which has the OutgoingTOS parameter with a binary value of 01000000 (2).

Then we started an FTP client on an AIX V4.3.2 system whose host name is rs600020.itso.ral.ibm.com. The FTP server is running on an OS/390 system named RA39.

During the FTP session, the trace data shown below had been taken by the tcpdump utility shipped with AIX V4:

```
root@rs600020[/]tcpdump -i tr0 -I host 172.16.232.39
tcpdump: (tr0):Promiscuous mode not supported for this interface.
        Continuing ...
tcpdump: listening on tr0
20:58:57.857023593 rs600020.itso.ral.ibm.com.55207 > 172.16.232.39.ftp: S
                        3590962606:3590962606(0) win 16384 <mss 512> [tos 0x10]
20:58:57.865927436 172.16.232.39.ftp > rs600020.itso.ral.ibm.com.55207: S
       2750408557:2750408557(0) ack 3590962607 win 32768 <mss 30668> [tos 0x40] 3
20:58:57.866424057 rs600020.itso.ral.ibm.com.55207 > 172.16.232.39.ftp: .
                        ack 1 win 16384 [tos 0x10]
20:59:00.134000802 172.16.232.39.ftp > rs600020.itso.ral.ibm.com.55207: P
                        1:64(63) ack 1 win 32768 [tos 0x40]  3
20:59:00.233136656 rs600020.itso.ral.ibm.com.55207 > 172.16.232.39.ftp: .
                        ack 64 win 16384 [tos 0x10]
20:59:00.240807436 172.16.232.39.ftp > rs600020.itso.ral.ibm.com.55207: P
                        64:125(61) ack 1 win 32768 [tos 0x40] 3
20:59:00.443219634 rs600020.itso.ral.ibm.com.55207 > 172.16.232.39.ftp: .
                        ack 125 win 16384 [tos 0x10]
```

Figure 182. Output from tcpdump with TOS = 40 (x'00100000')

3 The policy agent has updated the value of the TOS byte to the hexadecimal value of 40.

In the second test, the FTP ServicePolicyRules have been changed and the ServiceReference is now interactive1 (4) which has the OutgoingTOS parameter with a binary value of 10000000 (5).

```
ServicePolicyRules  ftpd            # FTP traffic
{
     ProtocolNumber     TCP
     Direction          Both
     SourcePortRange    20 21       # Both FTP control and data ports
     ServiceReference   interactive1      4
}

ServiceCategories   interactive1
{
     OutgoingTOS    10000000    5
}
```

Figure 183. Policy definition for FTP traffic - changing TOS value

Again we took a tcpdump data during an FTP session between an FTP client on AIX system and the FTP server on RA39.

The TOS byte has been set to 80 (6) as configured in the updated policy information:

```
root@rs600020[/]tcpdump -i tr0 -I host 172.16.232.39
tcpdump: (tr0):Promiscuous mode not supported for this interface.
        Continuing ...
tcpdump: listening on tr0
21:19:47.794038312 rs600020.itso.ral.ibm.com.55230 > 172.16.232.39.ftp: S
                        3749874606:3749874606(0) win 16384 <mss 512> [tos 0x10]
21:19:47.801728112 172.16.232.39.ftp > rs600020.itso.ral.ibm.com.55230: S
      3062894637:3062894637(0) ack 3749874607 win 32768 <mss 30668> [tos 0x80] 6
21:19:47.802226713 rs600020.itso.ral.ibm.com.55230 > 172.16.232.39.ftp: .
                        ack 1 win 16384 [tos 0x10]
21:19:47.872408473 172.16.232.39.ftp > rs600020.itso.ral.ibm.com.55230: P
                        1:64(63) ack 1 win 32768 [tos 0x80]    6
21:19:48.023174459 rs600020.itso.ral.ibm.com.55230 > 172.16.232.39.ftp: .
                        ack 64 win 16384 [tos 0x10]
21:19:48.030965299 172.16.232.39.ftp > rs600020.itso.ral.ibm.com.55230: P
                        64:125(61) ack 1 win 32768 [tos 0x80]  6
21:19:48.233160656 rs600020.itso.ral.ibm.com.55230 > 172.16.232.39.ftp: .
                        ack 125 win 16384 [tos 0x10]
```

Figure 184. Output from tcpdump with TOS = 40 (x'00100000')

Scenario 2

In this second scenario, we used the LDAP server to store the policies.

First from an FTP client on OS/390, we sent an MVS data set to an FTP server on an AIX system. The log from the FTP client looks like the following:

```
put bluepage bluepage.bin

 EZA1701I >>> SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=3200
 500 'SITE FIXRECFM 80 LRECL=80 RECFM=FB BLKSIZE=3200': command not understood.
 EZA1701I >>> PORT 172,16,232,39,4,19
 200 PORT command successful.
 EZA1701I >>> STOR bluepage.bin
 150 Opening data connection for bluepage.bin.
 226 Transfer complete.
 EZA1617I 4460160 bytes transferred in 6.900 seconds. Transfer rate 646.40 Kbytes,
 EZA1460I Command:
```

Figure 185. FTP client without MaxRate definition

The we defined the service policy for FTP as follows:

```
dn: s=srv39a2, o=IBM_US, c=US
objectclass: ServiceCategories
selectortag: MVS39A
servicename: srv39a2
tcpimagename: TCPIPA
policyscope: DataTraffic
maxrate: 400          1
minrate: 100          2
outgoingtos: 11100000

dn: r=rule39a2, o=IBM_US, c=US
objectclass: ServicePolicyRules
selectortag: MVS39A
policyrulesname: rule39a2
tcpimagename: TCPIPA
policyscope: Both
daysofweekmask: 1111110
timeofdayrange: 7:00-20:00
direction: Both
permission:Allowed
protocolnumber: TCP
DestinationPortRange: 20-21
servicereference: srv39a2
```

Figure 186. Service policies for FTP stored in LDAP server

We defined that MaxRate parameter as 400(Kbps) = 50(KBytes/sec) and the MinRate as 100(Kbps) = 12.5(KBytes/sec).

Then we loaded this service policy into the LDAP server using the ldapadd utility, and made sure that the policy had been installed in the TCP/IP stack. Then we ran the FTP client again.

```
put bluepage bluepage.bin

 EZA1701I >>> SITE FIXrecfm 80 LRECL=80 RECFM=FB BLKSIZE=3200
 500 'SITE FIXRECFM 80 LRECL=80 RECFM=FB BLKSIZE=3200': command not understood.
 EZA1701I >>> PORT 172,16,232,39,4,19
 200 PORT command successful.
 EZA1701I >>> STOR bluepage.bin
 150 Opening data connection for bluepage.bin.
 EZA1508I 800000 bytes transferred.
 EZA1508I 1440000 bytes transferred.
 EZA1508I 2080000 bytes transferred.
 EZA1508I 2720000 bytes transferred.
 EZA1508I 3360000 bytes transferred.
 EZA1508I 4000000 bytes transferred.
 226 Transfer complete.
 EZA1617I 4460160 bytes transferred in 81.700 seconds. Transfer rate 54.59 Kbps
```

Figure 187. FTP with MaxRate=400 and MinRate=100

As you can see in Figure 185 and Figure 187, the transfer rate in the second test case is a lot less than the first case. This test shows you how policies are applied. The policy agent tries to keep the transfer rate between the MaxRate and the MinRate.

B.3 Implement RSVP agent in CS for OS/390 V2R8 IP

The OS/390 RSVP agent is supported as a host RSVP implementation only. To run the RSVP agent in TCP/IP in CS OS/390 V2R8, we need to:

1. Configure the RSVPD configuration file that specifies RSVP agent operational parameters.
2. Start the RSVP agent.

B.3.1 RSVP configuration file

The RSVP configuration file can reside as either an MVS data set or HFS file. The RSVP agent uses the following search order to locate the configuration file:

1. HFS file or MVS data set specified in the -c start option
2. HFS file or MVS data set specified with the RSVPD_CONFIG_FILE environment variable
3. /etc/rsvpd.conf HFS file
4. hlq.RSVPD.CONF MVS data set

The statement used in the RSVP configuration file are:

- LogLevel statement, which specifies the level of tracing/logging
- TcpImage statement, which specifies the name of the stack to which the RSVP agent should establish affinity
- Interface statement, which makes one or more the network interfaces of the local host available to the process
- RSVP statement, which enables RSVP processing on one or more of the network interfaces of the local host

The sample of the configuration file is found in /etc/rsvpd.conf. Information on how to configure the file is available in the comments of the sample configuration file.

The following is our RSVPD configuration file on host MVS39A:

```
LogLevel 511
TcpImage TCPIPA                          1
Interface All                            2
{
   trafficcontrol enabled
}

RSVP All 3
```

Figure 188. RSVP configuration file

Figure 188 shows an RSVP configuration file defined in /etc/rsvpd.conf that is invoked by the RSVP started task:

1 TCPIPA is the stack name with which RSVP establishes an affinity.

2 **3** All interfaces defined in the HOME statement in the TCPIPA profile will be processed by RSVP. By configuring the `trafficcontrol` parameter in the Interface statement, traffic control is in effect. When traffic control is disabled, the RSVP agent does not install any filters (resource reservations).

B.3.2 RSVP startup

The RSVP can be started from the UNIX System Services shell or as a started task. At ITSO Raleigh, we used a started task procedure for the RSVP agent as follows:

```
//RSVPD    PROC
//*
//*
//RSVPD    EXEC PGM=RSVPD,REGION=0K,TIME=NOLIMIT,
//       PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/'
//*      PARM='POSIX(ON) ALL31(ON) ENVAR("_CEE_ENVFILE=DD:STDENV")/-c /
//*           etc/rsvpd.conf'
//*
//STDENV   DD DSN=TCPIP.TCPPARMS.R2612(RSVP&SYSCLONE.EV),DISP=SHR
//*STDENV   DD PATH='/etc/rsvpd2.env',PATHOPTS=(ORDONLY)
//*
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CEEDUMP  DD SYSOUT=*,DCB=(RECFM=FB,LRECL=132,BLKSIZE=132)
```

You can use environment variables, either defined in the data set or HFS file, specified by STDENV DD to run with the desired configuration. The following two variables can be configured:

- RESOLVER_CONFIG: specify TCP/IP configuration file
- RSVPD_CONFIG_FILE: specify RSVPD configuration file
- RSVPD_LOG_FILE: for the RSVPD log file

We put environment variables in TCPIP.TCPPARMS.R2612(RSVP39EV) which contains:

```
BROWSE    TCPIP.TCPPARMS.R2612(RSVP39EV) - 01.05     L
Command ===>
****************************** Top of Data *********
RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD39A)'
RSVPD_CONFIG_FILE=/etc/rsvpd.conf
RSVPD_LOG_FILE=/tmp/rsvpd.log

****************************** Bottom of Data *******
```

Note: Security Product (for example, RACF) authority is required to start the policy agent. The following commands can be used to create the necessary profile and permit users to use it:

```
RDEFINE OPERCMDS (MVS.SERVMGR.RSVPD) UACC(NONE)
PERMIT MVS.SERVMGR.RSVPD ACCESS(CONTROL) CLASS(OPERCMDS) ID(userid)
SETROPTS RACLIST(OPERCMDS) REFRESH
```

B.3.3 RSVP log file

To verify that RSVPD has been successfully initialized, we can check its log file which resides in /tmp/rsvpd.log. The following is the RSVPD log after RSVPD completely initialized.

```
INFO   :.main: *************** RSVP Agent started ***************
INFO   :...locate_configFile: Defaulted configuration file: /etc/rsvpd.conf
INFO   :.main: Using log level 511   1
INFO   :..settcpimage: Get TCP images rc - EDC8112I Operation not supported on socket.
INFO   :..settcpimage: Associate with TCP/IP image name = TCPIPA  2
INFO   :..reg_process: registering process with the system
INFO   :..reg_process: attempt OS/390 registration
INFO   :..reg_process: return from registration rc=0
TRACE  :...read_physical_netif: Home list entries returned = 7    3
INFO   :...read_physical_netif: index #0, interface VIPA39A has address 172.16.232.39, ifidx 0  4
INFO   :...read_physical_netif: index #1, interface TR1 has address 9.24.104.149, ifidx 1
INFO   :...read_physical_netif: index #2, interface M392216B has address 172.16.102.39, ifidx 2
INFO   :...read_physical_netif: index #3, interface EZAXCF28 has address 172.16.233.39, ifidx 3
WARNING:...read_physical_netif: interface 172.16.233.39 has been processed earlier
INFO   :...read_physical_netif: index #4, interface VIPLAC10F027 has address 172.16.240.39, ifidx 0
INFO   :...read_physical_netif: index #5, interface LOOPBACK has address 127.0.0.1, ifidx 0
INFO   :....mailslot_create: creating mailslot for timer
INFO   :...mailbox_register: mailbox allocated for timer
INFO   :.....mailslot_create: creating mailslot for RSVP
INFO   :....mailbox_register: mailbox allocated for rsvp
INFO   :.....mailslot_create: creating mailslot for RSVP via UDP
WARNING:.....mailslot_create: setsockopt(MCAST_ADD) failed - EDC5121I Invalid argument.
INFO   :....mailbox_register: mailbox allocated for rsvp-udp
TRACE  :..entity_initialize: interface 172.16.232.39, entity for rsvp allocated and initialized  5
INFO   :.....mailslot_create: creating mailslot for RSVP
INFO   :....mailbox_register: mailbox allocated for rsvp
INFO   :.....mailslot_create: creating mailslot for RSVP via UDP
INFO   :....mailbox_register: mailbox allocated for rsvp-udp
               :
INFO   :......mailslot_create: creating mailslot for route engine - informational socket
TRACE  :......mailslot_create: ready to accept informational socket connection
INFO   :.....mailbox_register: mailbox allocated for route
INFO   :.....mailslot_create: creating socket for traffic control module
INFO   :....mailbox_register: no mailbox necessary for traffic-control
INFO   :....mailslot_create: creating mailslot for RSVP client API
INFO   :...mailbox_register: mailbox allocated for rsvp-api
INFO   :...mailslot_create: creating mailslot for terminate
INFO   :..mailbox_register: mailbox allocated for terminate
INFO   :...mailslot_create: creating mailslot for dump
INFO   :..mailbox_register: mailbox allocated for dump
INFO   :...mailslot_create: creating mailslot for (broken) pipe
INFO   :..mailbox_register: mailbox allocated for pipe
INFO   :.main: rsvpd initialization complete    6
```

Figure 189. RSVP agent startup log message

1 With log level 511, all events are captured.

2 RSVP agent associates with TCPIPA.

3 RSVP processes seven interfaces that are currently activated in the HOME list of TCPIPA. You will see the list of interfaces in the NETSTAT HOME report. The RSVP agent processes all interfaces one by one (4, 5). Note that there are interfaces not defined in the HOME statement in TCPIP.PROFILE but activated dynamically, such as Dynamic XCF interfaces.

6 This message tells you the RSVP agent has been initialized successfully.

With the NETSTAT HOME report, you will see how RSVPD has established affinities with the TCP/IP stack.

```
D TCPIP,TCPIPA,N,CON
EZZ2500I NETSTAT CS V2R8 TCPIPA 787
USER ID  CONN     LOCAL SOCKET           FOREIGN SOCKET         STATE
BPXOINIT 00000024 0.0.0.0..10007         0.0.0.0..0             LISTEN
          :
RSVP1    00000192 127.0.0.1..1699        *..*                   UDP
RSVP1    0000018E 172.16.240.39..1699    *..*                   UDP
RSVP1    0000018C 172.16.233.39..1699    *..*                   UDP
RSVP1    0000018A 172.16.102.39..1699    *..*                   UDP
RSVP1    00000188 9.24.104.149..1699     *..*                   UDP
RSVP1    00000186 172.16.232.39..1699    *..*                   UDP
SYSLOGD1 00000036 0.0.0.0..514           *..*                   UDP
20 OF 20 RECORDS DISPLAYED
```

Figure 190. NETSTAT CONN report

B.4 RSVP API (RAPI)

In this section we discuss RSVP API, which allows you create an application to request networking resources explicitly.

Note that applications that were written to use the previous Web version of the RSVP agent have to be rebuilt because the RSVP API (RAPI) has changed from being statically linked to being shipped as a DLL.

B.4.1 RAPI overview

The OS/390 RSVP agent includes an application programming interface (API) for the Reservation Protocol (RSVP), known as RAPI. The RAPI interface is one realization of the generic API contained in the RSVP functional specification (refer to RFC 2205). The RAPI interface is a set of C language routines. With these APIs, you can create a custom application which requests enhanced quality of service (QoS). The RSVP agent then uses the RSVP protocol to propagate the QoS request through the routers along the paths for the data flow. Each router may accept or deny the request, depending upon the availability of resources. In the case of failure, the RSVP agent will return the decision to the requesting application using RAPI.

RSVP assigns QoS to specific IP data flows which can be either multipoint-to-multipoint or point-to-point data flows, known as sessions. A session is defined by a particular transport protocol, IP destination address, and destination port. To receive data packets for a particular multicast session, an application must join the corresponding IP multicast group.

Under RSVP, QoS requests are made by the data receivers. A QoS request contains a *flowspec*, together with a *filter spec*. The flowspec includes an Rspec (resource specification), which defines the desired QoS and is used to control the packet scheduling mechanism in the router or host, and also a Tspec (traffic specification), which defines the traffic expected by the receiver. The filter spec controls packet classification to determine which sender data packets receive the corresponding QoS.

The Tspec is composed of:

- r: Token bucket rate, which is the average data rate in bytes per second.
- b: Token bucket depth, which is the sending buffer in bytes.

- p: Peak data rate, in bytes per second.
- m: Nominal minimum packet size in bytes.
- M: Maximum packet size (MTU) in bytes.

The Rspec consists of the following values:

- R: The rate in bytes per second.
- S: The slack term in microseconds.

The following API calls are shipped with CS for OS/390 V2R8 IP:

- rapi_session: Establish an API session with the RSVP agent.
- rapi_release: Release an API session.
- rapi_sender: Indicate sender application. This call results in RSVP sending a PATH packet to the destination.
- rapi_reserve: Make a QoS reservation as a data receiver.
- rapi_dispatch, rapi_getfd: Support asynchronous upcall mechanism.

The following routines are standard routines for displaying the contents of RAPI objects.

- rapi_fmt_adspec: Formats a given RAPI Adspec into a buffer.
- rapi_fmt_filtspec: Formats a RAPI filter spec into a buffer.
- rapi_fmt_flowspec: Formats a given RAPI flowspec into a buffer.
- rapi_fmt_tspec: Formats a given RAPI Tspec into a buffer.

One or more RAPI sessions exist for a given RSVP session, each established with the rapi_session() call. Each sender within an RSVP session issues rapi_sender(), primarily to provide the sender Tspec. Each rapi_sender() call results in RSVP sending a PATH packet to the destination. The PATH packet is intercepted at each router along the data path and is used to install path state (the Tspec and other parameters).

When the PATH arrives at the destination, it is presented to the receiver application using the asynchronous upcall mechanism. The receiver uses the Tspec and other information to arrive at a reservation request, which is composed of one or more filter specs (which select one or more senders), and one or more flowspecs (which contain reservation parameters), based on the reservation style. These styles are supported:

- Wildcard style (WF): A single flowspec is shared among all current and future senders.
- Fixed style (FF): A specific flowspec is paired with a specific filter spec for a specific sender. One or more pairs may be specified.
- Shared Explicit style (SE): A single flowspec is shared among a named set of senders.

The reservation parameters take two forms, depending on the service type being used. The following service types are supported:

- Controlled Load (CL): The reservation parameters take the same form as the Tspec (r, b, p, m, M). The goal of CL service is to provide what appears to be unloaded service response characteristics even when one or more hosts or routers along the path is heavily loaded.

- Guaranteed (GUAR): The reservation parameters take the form of the Tspec (r, b, p, m, M) followed by an Rspec (R, S). The additional Rspec parameters provide a requested rate in bytes per second, and a slack term in microseconds. The goal of GUAR service is to use the various parameters in specific equations to produce the maximum delay that will be experienced when sending data. By changing certain parameters, the application can control, to a certain extent, the value of that delay.

The receiver application makes the reservation request using the rapi_reserve() call. This causes RSVP to send an RESV packet hop-by-hop back along the same path traveled by the PATH packet. Each host or router along the path installs the appropriate reservation on the appropriate interface.

When the RESV arrives at the sender, it is presented asynchronously to the sender application. The application may or may not choose to wait for the RESV before commencing data transmission, however, all data sent before the reservation is in place is delivered best effort.

The upcall mechanism requires the application to use a listener thread, which gets a socket descriptor to use via the rapi_getfd() call, then uses this socket descriptor in a select() call. When an RSVP packet arrives, the select() will return, at which point the application uses rapi_dispatch() to cause the asynchronous function registered on the rapi_session() call to receive control. All RAPI/RSVP objects, such as Tspecs and flowspecs, are delivered to this function as parameters.

An application using RAPI must compile with the <rapi.h> header file (available in /usr/include) and use the DLL compiler option.

If the prelinker is used, then the application must be prelinked with the rapi.x definition side deck, which is available in /usr/lib. If the Binder is used instead of the prelinker, then the rapi.x file must be included with the application object files, and the Binder DLL option must be specified.

The application must have access to the rapi.dll, available in /usr/lib, at run time. This is usually accomplished by using the LIBPATH environment variable.

Appendix C. Management Information Base (MIB) objects

This appendix lists the objects defined by the Management Information Base (MIB) that are supported by the SNMP agent and subagents on the CS for OS/390 and the maximum access allowed.

Note: If an SNMP SET (write) is attempted against a variable for which the maximum access is read-only, an error code is returned. For an SNMPv2 request, the error code is noAccess or notWritable.

The object types are defined using the following fields:

- Object descriptor

  A textual name for the object type, along with its corresponding object identifier.

- Object identifier

  The name for the object type, using ASN.1 notation.

- Supported by

  Support by the agent or subagents. If support is by one of the subagents, the subagent is named. Supported subagents include:

  - TCP/IP
  - OMPRoute
  - SLA

- Defined by

  The location of the description of the object. The SNMP agent provides support of the following enterprise-specific MIBs:

  - Subagent MIB
  - Extensions to the DPI20 MIB defined by RFC 1592

  The TCP/IP subagent provides support of the following enterprise-specific MIBs:

  - IBM 3172 MIB
  - IBM TCP/IP MVS Enterprise Specific MIB (which includes remote ping)

  Copies of the SMI syntax for the above-mentioned MIBs are installed in the HFS directory /usr/lpp/tcpip/samples as:

  - mvstcpip.mi2 (SNMPv2 SMI)
  - mvstcpip.mib (SNMPv1 SMI)
  - saMIB.mi2 (SNMPv2 SMI)
  - saMIB.mib (SNMPv1 SMI)
  - slapm.mi2 (SNMPv2 SMI)
  - slapm.mib (SNMPv1 SMI)
  - slapm.txt (draft RFC)
  - rfc1592b.mi2 (SNMPv2 SMI)
  - rfc1592b.mib (SNMPv1 SMI)
  - ibm3172.mi2 (SNMPv2 SMI)
  - ibm3172.mib (SNMPv1 SMI)

- AA - Access Allowed

  - Read-only (R/O)

- Read-write (R/W)
- Read-create (R/C)
- Write-only (W/O)
- Not-accessible (N/A)

Table 26. Management Information Base (MIB) Objects

Object Descriptor	Object Identifier	Supported by	Defined by	AA
sysDescr	1.3.6.1.2.1.1.1	Agent	RFC1907	R/O
sysObjectID	1.3.6.1.2.1.1.2	Agent	RFC1907	R/O
sysUpTime	1.3.6.1.2.1.1.3	Agent	RFC1907	R/O
sysContact	1.3.6.1.2.1.1.4	Agent	RFC1907	R/W
sysName	1.3.6.1.2.1.1.5	Agent	RFC1907	R/W
sysLocation	1.3.6.1.2.1.1.6	Agent	RFC1907	R/W
sysServices	1.3.6.1.2.1.1.7	Agent	RFC1907	R/O
sysORLastChange	1.3.6.1.2.1.1.8	Agent	RFC1907	R/O
sysORTable	1.3.6.1.2.1.1.9	Agent	RFC1907	N/A
sysOREntry	1.3.6.1.2.1.1.9.1	Agent	RFC1907	N/A
sysORIndex	1.3.6.1.2.1.1.9.1.1	Agent	RFC1907	N/A
sysORID	1.3.6.1.2.1.1.9.1.2	Agent	RFC1907	R/O
sysORDescr	1.3.6.1.2.1.1.9.1.3	Agent	RFC1907	R/O
sysORUpTime	1.3.6.1.2.1.1.9.1.4	Agent	RFC1907	R/O
ifTable	1.3.6.1.2.1.2.2	TCP/IP	RFC2233	N/A
ifEntry	1.3.6.1.2.1.2.2.1	TCP/IP	RFC2233	N/A
ifIndex	1.3.6.1.2.1.2.2.1.1	TCP/IP	RFC2233	R/O
ifDescr	1.3.6.1.2.1.2.2.1.2	TCP/IP	RFC2233	R/O
ifType	1.3.6.1.2.1.2.2.1.3	TCP/IP	RFC2233	R/O
ifMtu	1.3.6.1.2.1.2.2.1.4	TCP/IP	RFC2233	R/O
ifSpeed	1.3.6.1.2.1.2.2.1.5	TCP/IP	RFC2233	R/O
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	TCP/IP	RFC2233	R/O
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	TCP/IP	RFC2233	R/W
ifOperStatus	1.3.6.1.2.1.2.2.1.8	TCP/IP	RFC2233	R/O
ifLastChange	1.3.6.1.2.1.2.2.1.9	TCP/IP	RFC2233	R/O
ifInOctets	1.3.6.1.2.1.2.2.1.10	TCP/IP	RFC2233	R/O
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	TCP/IP	RFC2233	R/O
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	TCP/IP	RFC2233	R/O
ifInDiscards	1.3.6.1.2.1.2.2.1.13	TCP/IP	RFC2233	R/O
ifInErrors	1.3.6.1.2.1.2.2.1.14	TCP/IP	RFC2233	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	TCP/IP	RFC2233	R/O
ifOutOctets	1.3.6.1.2.1.2.2.1.16	TCP/IP	RFC2233	R/O
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	TCP/IP	RFC2233	R/O
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	TCP/IP	RFC2233	R/O
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	TCP/IP	RFC2233	R/O
ifOutErrors	1.3.6.1.2.1.2.2.1.20	TCP/IP	RFC2233	R/O
ifOutQLen	1.3.6.1.2.1.2.2.1.21	TCP/IP	RFC2233	R/O
ifSpecific	1.3.6.1.2.1.2.2.1.22	TCP/IP	RFC2233	R/O
ipForwarding	1.3.6.1.2.1.4.1	TCP/IP	RFC2011	R/W
ipDefaultTTL	1.3.6.1.2.1.4.2	TCP/IP	RFC2011	R/W
ipInReceives	1.3.6.1.2.1.4.3	TCP/IP	RFC2011	R/O
ipInHdrErrors	1.3.6.1.2.1.4.4	TCP/IP	RFC2011	R/O
ipInAddrErrors	1.3.6.1.2.1.4.5	TCP/IP	RFC2011	R/O
ipForwDatagrams	1.3.6.1.2.1.4.6	TCP/IP	RFC2011	R/O
ipInUnknownProtos	1.3.6.1.2.1.4.7	TCP/IP	RFC2011	R/O
ipInDiscards	1.3.6.1.2.1.4.8	TCP/IP	RFC2011	R/O
ipInDelivers	1.3.6.1.2.1.4.9	TCP/IP	RFC2011	R/O
ipOutRequests	1.3.6.1.2.1.4.10	TCP/IP	RFC2011	R/O
ipOutDiscards	1.3.6.1.2.1.4.11	TCP/IP	RFC2011	R/O
ipOutNoRoutes	1.3.6.1.2.1.4.12	TCP/IP	RFC2011	R/O
ipReasmTimeout	1.3.6.1.2.1.4.13	TCP/IP	RFC2011	R/W
ipReasmReqds	1.3.6.1.2.1.4.14	TCP/IP	RFC2011	R/O
ipReasmOKs	1.3.6.1.2.1.4.15	TCP/IP	RFC2011	R/O
ipReasmFails	1.3.6.1.2.1.4.16	TCP/IP	RFC2011	R/O
ipFragOKs	1.3.6.1.2.1.4.17	TCP/IP	RFC2011	R/O
ipFragFails	1.3.6.1.2.1.4.18	TCP/IP	RFC2011	R/O
ipFragCreates	1.3.6.1.2.1.4.19	TCP/IP	RFC2011	R/O
ipAddrTable	1.3.6.1.2.1.4.20	TCP/IP	RFC2011	N/A
ipAddrEntry	1.3.6.1.2.1.4.20.1	TCP/IP	RFC2011	N/A
ipAdEntAddr	1.3.6.1.2.1.4.20.1.1	TCP/IP	RFC2011	R/O
ipAdEntIfIndex	1.3.6.1.2.1.4.20.1.2	TCP/IP	RFC2011	R/O
ipAdEntNetMask	1.3.6.1.2.1.4.20.1.3	TCP/IP	RFC2011	R/O
ipAdEntBcastAddr	1.3.6.1.2.1.4.20.1.4	TCP/IP	RFC2011	R/O
ipAdEntReasmMaxSize	1.3.6.1.2.1.4.20.1.5	TCP/IP	RFC2011	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ipNetToMediaTable	1.3.6.1.2.1.4.22	TCP/IP	RFC2011	N/A
ipNetToMediaEntry	1.3.6.1.2.1.4.22.1	TCP/IP	RFC2011	N/A
ipNetToMediaIfIndex	1.3.6.1.2.1.4.22.1.1	TCP/IP	RFC2011	R/O
ipNetToMediaPhysAddress	1.3.6.1.2.1.4.22.1.2	TCP/IP	RFC2011	R/O
ipNetToMediaNetAddress	1.3.6.1.2.1.4.22.1.3	TCP/IP	RFC2011	R/O
ipNetToMediaType	1.3.6.1.2.1.4.22.1.4	TCP/IP	RFC2011	R/O
ipRoutingDiscards	1.3.6.1.2.1.4.23	TCP/IP	RFC2011	R/O
ipForward	1.3.6.1.2.1.4.24	TCP/IP	RFC1354	N/A
ipForwardNumber	1.3.6.1.2.1.4.24.1	TCP/IP	RFC1354	R/O
ipForwardTable	1.3.6.1.2.1.4.24.2	TCP/IP	RFC1354	N/A
ipForwardEntry	1.3.6.1.2.1.4.24.2.1	TCP/IP	RFC1354	N/A
ipForwardDest	1.3.6.1.2.1.4.24.2.1.1	TCP/IP	RFC1354	R/O
ipForwardMask	1.3.6.1.2.1.4.24.2.1.2	TCP/IP	RFC1354	R/O
ipForwardPolicy	1.3.6.1.2.1.4.24.2.1.3	TCP/IP	RFC1354	R/O
ipForwardNextHop	1.3.6.1.2.1.4.24.2.1.4	TCP/IP	RFC1354	R/O
ipForwardIfIndex	1.3.6.1.2.1.4.24.2.1.5	TCP/IP	RFC1354	R/O
ipForwardType	1.3.6.1.2.1.4.24.2.1.6	TCP/IP	RFC1354	R/O
ipForwardProto	1.3.6.1.2.1.4.24.2.1.7	TCP/IP	RFC1354	R/O
ipForwardAge	1.3.6.1.2.1.4.24.2.1.8	TCP/IP	RFC1354	R/O
ipForwardInfo	1.3.6.1.2.1.4.24.2.1.9	TCP/IP	RFC1354	R/O
ipForwardNextHopAS	1.3.6.1.2.1.4.24.2.1.10	TCP/IP	RFC1354	R/O
ipForwardMetric1	1.3.6.1.2.1.4.24.2.1.11	TCP/IP	RFC1354	R/O
ipForwardMetric2	1.3.6.1.2.1.4.24.2.1.12	TCP/IP	RFC1354	R/O
ipForwardMetric3	1.3.6.1.2.1.4.24.2.1.13	TCP/IP	RFC1354	R/O
ipForwardMetric4	1.3.6.1.2.1.4.24.2.1.14	TCP/IP	RFC1354	R/O
ipForwardMetric5	1.3.6.1.2.1.4.24.2.1.15	TCP/IP	RFC1354	R/O
icmpInMsgs	1.3.6.1.2.1.5.1	TCP/IP	RFC2011	R/O
icmpInErrors	1.3.6.1.2.1.5.2	TCP/IP	RFC2011	R/O
icmpInDestUnreachs	1.3.6.1.2.1.5.3	TCP/IP	RFC2011	R/O
icmpInTimeExcds	1.3.6.1.2.1.5.4	TCP/IP	RFC2011	R/O
icmpInParmProbs	1.3.6.1.2.1.5.5	TCP/IP	RFC2011	R/O
icmpInSrcQuenchs	1.3.6.1.2.1.5.6	TCP/IP	RFC2011	R/O
icmpInRedirects	1.3.6.1.2.1.5.7	TCP/IP	RFC2011	R/O
icmpInEchos	1.3.6.1.2.1.5.8	TCP/IP	RFC2011	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
icmpInEchoReps	1.3.6.1.2.1.5.9	TCP/IP	RFC2011	R/O
icmpInTimestamps	1.3.6.1.2.1.5.10	TCP/IP	RFC2011	R/O
icmpInTimestampReps	1.3.6.1.2.1.5.11	TCP/IP	RFC2011	R/O
icmpInAddrMasks	1.3.6.1.2.1.5.12	TCP/IP	RFC2011	R/O
icmpInAddrMaskReps	1.3.6.1.2.1.5.13	TCP/IP	RFC2011	R/O
icmpOutMsgs	1.3.6.1.2.1.5.14	TCP/IP	RFC2011	R/O
icmpOutErrors	1.3.6.1.2.1.5.15	TCP/IP	RFC2011	R/O
icmpOutDestUnreachs	1.3.6.1.2.1.5.16	TCP/IP	RFC2011	R/O
icmpOutTimeExcds	1.3.6.1.2.1.5.17	TCP/IP	RFC2011	R/O
icmpOutParmProbs	1.3.6.1.2.1.5.18	TCP/IP	RFC2011	R/O
icmpOutSrcQuenchs	1.3.6.1.2.1.5.19	TCP/IP	RFC2011	R/O
icmpOutRedirects	1.3.6.1.2.1.5.20	TCP/IP	RFC2011	R/O
icmpOutEchos	1.3.6.1.2.1.5.21	TCP/IP	RFC2011	R/O
icmpOutEchoReps	1.3.6.1.2.1.5.22	TCP/IP	RFC2011	R/O
icmpOutTimestamps	1.3.6.1.2.1.5.23	TCP/IP	RFC2011	R/O
icmpOutTimestampReps	1.3.6.1.2.1.5.24	TCP/IP	RFC2011	R/O
icmpOutAddrMasks	1.3.6.1.2.1.5.25	TCP/IP	RFC2011	R/O
icmpOutAddrMaskReps	1.3.6.1.2.1.5.26	TCP/IP	RFC2011	R/O
tcpRtoAlgorithm	1.3.6.1.2.1.6.1	TCP/IP	RFC2012	R/O
tcpRtoMin	1.3.6.1.2.1.6.2	TCP/IP	RFC2012	R/O
tcpRtoMax	1.3.6.1.2.1.6.3	TCP/IP	RFC2012	R/O
tcpMaxConn	1.3.6.1.2.1.6.4	TCP/IP	RFC2012	R/O
tcpActiveOpens	1.3.6.1.2.1.6.5	TCP/IP	RFC2012	R/O
tcpPassiveOpens	1.3.6.1.2.1.6.6	TCP/IP	RFC2012	R/O
tcpAttemptFails	1.3.6.1.2.1.6.7	TCP/IP	RFC2012	R/O
tcpEstabResets	1.3.6.1.2.1.6.8	TCP/IP	RFC2012	R/O
tcpCurrEstab	1.3.6.1.2.1.6.9	TCP/IP	RFC2012	R/O
tcpInSegs	1.3.6.1.2.1.6.10	TCP/IP	RFC2012	R/O
tcpOutSegs	1.3.6.1.2.1.6.11	TCP/IP	RFC2012	R/O
tcpRetransSegs	1.3.6.1.2.1.6.12	TCP/IP	RFC2012	R/O
tcpConnTable	1.3.6.1.2.1.6.13	TCP/IP	RFC2012	N/A
tcpConnEntry	1.3.6.1.2.1.6.13.1	TCP/IP	RFC2012	N/A
tcpConnState	1.3.6.1.2.1.6.13.1.1	TCP/IP	RFC2012	R/W
tcpConnLocalAddress	1.3.6.1.2.1.6.13.1.2	TCP/IP	RFC2012	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
tcpConnLocalPort	1.3.6.1.2.1.6.13.1.3	TCP/IP	RFC2012	R/O
tcpConnRemAddress	1.3.6.1.2.1.6.13.1.4	TCP/IP	RFC2012	R/O
tcpConnRemPort	1.3.6.1.2.1.6.13.1.5	TCP/IP	RFC2012	R/O
tcpInErrs	1.3.6.1.2.1.6.14	TCP/IP	RFC2012	R/O
tcpOutRsts	1.3.6.1.2.1.6.15	TCP/IP	RFC2012	R/O
udpInDatagrams	1.3.6.1.2.1.7.1	TCP/IP	RFC2013	R/O
udpNoPorts	1.3.6.1.2.1.7.2	TCP/IP	RFC2013	R/O
udpInErrors	1.3.6.1.2.1.7.3	TCP/IP	RFC2013	R/O
udpOutDatagrams	1.3.6.1.2.1.7.4	TCP/IP	RFC2013	R/O
udpTable	1.3.6.1.2.1.7.5	TCP/IP	RFC2013	N/A
udpEntry	1.3.6.1.2.1.7.5.1	TCP/IP	RFC2013	N/A
udpLocalAddress	1.3.6.1.2.1.7.5.1.1	TCP/IP	RFC2013	R/O
udpLocalPort	1.3.6.1.2.1.7.5.1.2	TCP/IP	RFC2013	R/O
snmpInPkts	1.3.6.1.2.1.11.1	Agent	RFC1907	R/O
snmpInBadVersions	1.3.6.1.2.1.11.3	Agent	RFC1907	R/O
snmpInBadCommunityNames	1.3.6.1.2.1.11.4	Agent	RFC1907	R/O
snmpInBadCommunityUses	1.3.6.1.2.1.11.5	Agent	RFC1907	R/O
snmpInASNParseErrs	1.3.6.1.2.1.11.6	Agent	RFC1907	R/O
snmpEnableAuthenTraps	1.3.6.1.2.1.11.30	Agent	RFC1907	R/W
snmpSilentDrops	1.3.6.1.2.1.11.31	Agent	RFC1907	R/O
snmpProxyDrops	1.3.6.1.2.1.11.32	Agent	RFC1907	R/O
ospf	1.3.6.1.2.1.14	omproute	RFC1850	N/A
ospfGeneralGroup	1.3.6.1.2.1.14.1	omproute	RFC1850	N/A
ospfRouterId	1.3.6.1.2.1.14.1.1	omproute	RFC1850	R/O
ospfAdminStat	1.3.6.1.2.1.14.1.2	omproute	RFC1850	R/O
ospfVersionNumber	1.3.6.1.2.1.14.1.3	omproute	RFC1850	R/O
ospfAreaBdrRtrStatus	1.3.6.1.2.1.14.1.4	omproute	RFC1850	R/O
ospfASBdrRtrStatus	1.3.6.1.2.1.14.1.5	omproute	RFC1850	R/O
ospfExternLsaCount	1.3.6.1.2.1.14.1.6	omproute	RFC1850	R/O
ospfExternLsaCksumSum	1.3.6.1.2.1.14.1.7	omproute	RFC1850	R/O
ospfTOSSupport	1.3.6.1.2.1.14.1.8	omproute	RFC1850	R/O
ospfOriginateNewLsas	1.3.6.1.2.1.14.1.9	omproute	RFC1850	R/O
ospfRxNewLsas	1.3.6.1.2.1.14.1.10	omproute	RFC1850	R/O
ospfExtLsdbLimit	1.3.6.1.2.1.14.1.11	omproute	RFC1850	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ospfMulticastExtensions	1.3.6.1.2.1.14.1.12	omproute	RFC1850	R/O
ospfDemandExtensions	1.3.6.1.2.1.14.1.14	omproute	RFC1850	R/O
ospfAreaTable	1.3.6.1.2.1.14.2	omproute	RFC1850	N/A
ospfAreaEntry	1.3.6.1.2.1.14.2.1	omproute	RFC1850	N/A
ospfAreaId	1.3.6.1.2.1.14.2.1.1	omproute	RFC1850	R/O
ospfImportAsExtern	1.3.6.1.2.1.14.2.1.3	omproute	RFC1850	R/O
ospfSpfRuns	1.3.6.1.2.1.14.2.1.4	omproute	RFC1850	R/O
ospfAreaBdrRtrCount	1.3.6.1.2.1.14.2.1.5	omproute	RFC1850	R/O
ospfAsBdrRtrCount	1.3.6.1.2.1.14.2.1.6	omproute	RFC1850	R/O
ospfAreaLsaCount	1.3.6.1.2.1.14.2.1.7	omproute	RFC1850	R/O
ospfAreaLsaCksumSum	1.3.6.1.2.1.14.2.1.8	omproute	RFC1850	R/O
ospfAreaSummary	1.3.6.1.2.1.14.2.1.9	omproute	RFC1850	R/O
ospfStubAreaTable	1.3.6.1.2.1.14.3	omproute	RFC1850	N/A
ospfStubAreaEntry	1.3.6.1.2.1.14.3.1	omproute	RFC1850	N/A
ospfStubAreaId	1.3.6.1.2.1.14.3.1.1	omproute	RFC1850	R/O
ospfStubTOS	1.3.6.1.2.1.14.3.1.2	omproute	RFC1850	R/O
ospfStubMetric	1.3.6.1.2.1.14.3.1.3	omproute	RFC1850	R/O
ospfStubMetricType	1.3.6.1.2.1.14.3.1.5	omproute	RFC1850	R/O
ospfLsdbTable	1.3.6.1.2.1.14.4	omproute	RFC1850	N/A
ospfLsdbEntry	1.3.6.1.2.1.14.4.1	omproute	RFC1850	N/A
ospfLsdbAreaId	1.3.6.1.2.1.14.4.1.1	omproute	RFC1850	R/O
ospfLsdbType	1.3.6.1.2.1.14.4.1.2	omproute	RFC1850	R/O
ospfLsdbLsid	1.3.6.1.2.1.14.4.1.3	omproute	RFC1850	R/O
ospfLsdbRouterId	1.3.6.1.2.1.14.4.1.4	omproute	RFC1850	R/O
ospfLsdbSequence	1.3.6.1.2.1.14.4.1.5	omproute	RFC1850	R/O
ospfLsdbAge	1.3.6.1.2.1.14.4.1.6	omproute	RFC1850	R/O
ospfLsdbChecksum	1.3.6.1.2.1.14.4.1.7	omproute	RFC1850	R/O
ospfLsdbAdvertisement	1.3.6.1.2.1.14.4.1.8	omproute	RFC1850	R/O
ospfIfTable	1.3.6.1.2.1.14.7	omproute	RFC1850	N/A
ospfIfEntry	1.3.6.1.2.1.14.7.1	omproute	RFC1850	N/A
ospfIfIpAddress	1.3.6.1.2.1.14.7.1.1	omproute	RFC1850	R/O
ospfAddressLessIf	1.3.6.1.2.1.14.7.1.2	omproute	RFC1850	R/O
ospfIfAreaId	1.3.6.1.2.1.14.7.1.3	omproute	RFC1850	R/O
ospfIfType	1.3.6.1.2.1.14.7.1.4	omproute	RFC1850	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ospfIfAdminStat	1.3.6.1.2.1.14.7.1.5	omproute	RFC1850	R/O
ospfIfRtrPriority	1.3.6.1.2.1.14.7.1.6	omproute	RFC1850	R/O
ospfIfTransitDelay	1.3.6.1.2.1.14.7.1.7	omproute	RFC1850	R/O
ospfIfRetransInterval	1.3.6.1.2.1.14.7.1.8	omproute	RFC1850	R/O
ospfIfHelloInterval	1.3.6.1.2.1.14.7.1.9	omproute	RFC1850	R/O
ospfIfRtrDeadInterval	1.3.6.1.2.1.14.7.1.10	omproute	RFC1850	R/O
ospfIfPollInterval	1.3.6.1.2.1.14.7.1.11	omproute	RFC1850	R/O
ospfIfState	1.3.6.1.2.1.14.7.1.12	omproute	RFC1850	R/O
ospfIfDesignatedRouter	1.3.6.1.2.1.14.7.1.13	omproute	RFC1850	R/O
ospfIfBackupDesignatedRouter	1.3.6.1.2.1.14.7.1.14	omproute	RFC1850	R/O
ospfIfEvents	1.3.6.1.2.1.14.7.1.15	omproute	RFC1850	R/O
ospfIfAuthKey	1.3.6.1.2.1.14.7.1.16	omproute	RFC1850	R/O
ospfIfMulticastForwarding	1.3.6.1.2.1.14.7.1.18	omproute	RFC1850	R/O
ospfIfDemand	1.3.6.1.2.1.14.7.1.19	omproute	RFC1850	R/O
ospfIfAuthType	1.3.6.1.2.1.14.7.1.20	omproute	RFC1850	R/O
ospfIfMetricTable	1.3.6.1.2.1.14.8	omproute	RFC1850	N/A
ospfIfMetricEntry	1.3.6.1.2.1.14.8.1	omproute	RFC1850	N/A
ospfIfMetricIpAddress	1.3.6.1.2.1.14.8.1.1	omproute	RFC1850	R/O
ospfIfMetricAddressLessIf	1.3.6.1.2.1.14.8.1.2	omproute	RFC1850	R/O
ospfIfMetricTOS	1.3.6.1.2.1.14.8.1.3	omproute	RFC1850	R/O
ospfIfMetricValue	1.3.6.1.2.1.14.8.1.4	omproute	RFC1850	R/O
ospfVirtIfTable	1.3.6.1.2.1.14.9	omproute	RFC1850	N/A
ospfVirtIfEntry	1.3.6.1.2.1.14.9.1	omproute	RFC1850	N/A
ospfVirtIfAreaId	1.3.6.1.2.1.14.9.1.1	omproute	RFC1850	R/O
ospfVirtIfNeighbor	1.3.6.1.2.1.14.9.1.2	omproute	RFC1850	R/O
ospfVirtIfTransitDelay	1.3.6.1.2.1.14.9.1.3	omproute	RFC1850	R/O
ospfVirtIfRetransInterval	1.3.6.1.2.1.14.9.1.4	omproute	RFC1850	R/O
ospfVirtIfHelloInterval	1.3.6.1.2.1.14.9.1.5	omproute	RFC1850	R/O
ospfVirtIfRtrDeadInterval	1.3.6.1.2.1.14.9.1.6	omproute	RFC1850	R/O
ospfVirtIfState	1.3.6.1.2.1.14.9.1.7	omproute	RFC1850	R/O
ospfVirtIfEvents	1.3.6.1.2.1.14.9.1.8	omproute	RFC1850	R/O
ospfVirtIfAuthKey	1.3.6.1.2.1.14.9.1.9	omproute	RFC1850	R/O
ospfVirtIfAuthType	1.3.6.1.2.1.14.9.1.11	omproute	RFC1850	R/O
ospfNbrTable	1.3.6.1.2.1.14.10	omproute	RFC1850	N/A

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ospfNbrEntry	1.3.6.1.2.1.14.10.1	omproute	RFC1850	N/A
ospfNbrIpAddr	1.3.6.1.2.1.14.10.1.1	omproute	RFC1850	R/O
ospfNbrAddressLessIndex	1.3.6.1.2.1.14.10.1.2	omproute	RFC1850	R/O
ospfNbrRtrId	1.3.6.1.2.1.14.10.1.3	omproute	RFC1850	R/O
ospfNbrOptions	1.3.6.1.2.1.14.10.1.4	omproute	RFC1850	R/O
ospfNbrPriority	1.3.6.1.2.1.14.10.1.5	omproute	RFC1850	R/O
ospfNbrState	1.3.6.1.2.1.14.10.1.6	omproute	RFC1850	R/O
ospfNbrEvents	1.3.6.1.2.1.14.10.1.7	omproute	RFC1850	R/O
ospfNbrLsRetransQLen	1.3.6.1.2.1.14.10.1.8	omproute	RFC1850	R/O
ospfNbmaNbrPermanence	1.3.6.1.2.1.14.10.1.10	omproute	RFC1850	R/O
ospfNbrHelloSuppressed	1.3.6.1.2.1.14.10.1.11	omproute	RFC1850	R/O
ospfVirtNbrTable	1.3.6.1.2.1.14.11	omproute	RFC1850	N/A
ospfVirtNbrEntry	1.3.6.1.2.1.14.11.1	omproute	RFC1850	N/A
ospfVirtNbrArea	1.3.6.1.2.1.14.11.1.1	omproute	RFC1850	R/O
ospfVirtNbrRtrId	1.3.6.1.2.1.14.11.1.2	omproute	RFC1850	R/O
ospfVirtNbrIpAddr	1.3.6.1.2.1.14.11.1.3	omproute	RFC1850	R/O
ospfVirtNbrOptions	1.3.6.1.2.1.14.11.1.4	omproute	RFC1850	R/O
ospfVirtNbrState	1.3.6.1.2.1.14.11.1.5	omproute	RFC1850	R/O
ospfVirtNbrEvents	1.3.6.1.2.1.14.11.1.6	omproute	RFC1850	R/O
ospfVirtNbrLsRetransQLen	1.3.6.1.2.1.14.11.1.7	omproute	RFC1850	R/O
ospfVirtNbrHelloSuppressed	1.3.6.1.2.1.14.11.1.8	omproute	RFC1850	R/O
ospfExtLsdbTable	1.3.6.1.2.1.14.12	omproute	RFC1850	N/A
ospfExtLsdbEntry	1.3.6.1.2.1.14.12.1	omproute	RFC1850	N/A
ospfExtLsdbType	1.3.6.1.2.1.14.12.1.1	omproute	RFC1850	R/O
ospfExtLsdbLsid	1.3.6.1.2.1.14.12.1.2	omproute	RFC1850	R/O
ospfExtLsdbRouterId	1.3.6.1.2.1.14.12.1.3	omproute	RFC1850	R/O
ospfExtLsdbSequence	1.3.6.1.2.1.14.12.1.4	omproute	RFC1850	R/O
ospfExtLsdbAge	1.3.6.1.2.1.14.12.1.5	omproute	RFC1850	R/O
ospfExtLsdbChecksum	1.3.6.1.2.1.14.12.1.6	omproute	RFC1850	R/O
ospfExtLsdbAdvertisement	1.3.6.1.2.1.14.12.1.7	omproute	RFC1850	R/O
ospfAreaAggregateTable	1.3.6.1.2.1.14.14	omproute	RFC1850	N/A
ospfAreaAggregateEntry	1.3.6.1.2.1.14.14.1	omproute	RFC1850	N/A
ospfAreaAggregateAreaID	1.3.6.1.2.1.14.14.1.1	omproute	RFC1850	R/O
ospfAreaAggregateLsdbType	1.3.6.1.2.1.14.14.1.2	omproute	RFC1850	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ospfAreaAggregateNet	1.3.6.1.2.1.14.14.1.3	omproute	RFC1850	R/O
ospfAreaAggregateMask	1.3.6.1.2.1.14.14.1.4	omproute	RFC1850	R/O
ospfAreaAggregateEffect	1.3.6.1.2.1.14.14.1.6	omproute	RFC1850	R/O
ifXTable	1.3.6.1.2.1.31.1.1	TCP/IP	RFC2233	N/A
ifXEntry	1.3.6.1.2.1.31.1.1.1	TCP/IP	RFC2233	N/A
ifName	1.3.6.1.2.1.31.1.1.1.1	TCP/IP	RFC2233	R/O
ifInMulticastPkts	1.3.6.1.2.1.31.1.1.1.2	TCP/IP	RFC2233	R/O
ifInBroadcastPkts	1.3.6.1.2.1.31.1.1.1.3	TCP/IP	RFC2233	R/O
ifOutMulticastPkts	1.3.6.1.2.1.31.1.1.1.4	TCP/IP	RFC2233	R/O
ifOutBroadcastPkts	1.3.6.1.2.1.31.1.1.1.5	TCP/IP	RFC2233	R/O
ifHCInOctets	1.3.6.1.2.1.31.1.1.1.6	TCP/IP	RFC2233	R/O
ifHCOutOctets	1.3.6.1.2.1.31.1.1.1.10	TCP/IP	RFC2233	R/O
ifLinkUpDownTrapEnable	1.3.6.1.2.1.31.1.1.1.14	TCP/IP	RFC2233	R/W
ifHighSpeed	1.3.6.1.2.1.31.1.1.1.15	TCP/IP	RFC2233	R/O
ifPromiscuousMode	1.3.6.1.2.1.31.1.1.1.16	TCP/IP	RFC2233	R/O
ifConnectorPresent	1.3.6.1.2.1.31.1.1.1.17	TCP/IP	RFC2233	R/O
ifAlias	1.3.6.1.2.1.31.1.1.1.18	TCP/IP	RFC2233	R/W
ifCounterDiscontinuityTime	1.3.6.1.2.1.31.1.1.1.19	TCP/IP	RFC2233	R/O
ifStackTable	1.3.6.1.2.1.31.1.2	TCP/IP	RFC2233	N/A
ifStackEntry	1.3.6.1.2.1.31.1.2.1	TCP/IP	RFC2233	N/A
ifStackStatus	1.3.6.1.2.1.31.1.2.1.3	TCP/IP	RFC2233	R/O
ipoaLisTable	1.3.6.1.2.1.10.46.1.2.	TCP/IP	RFC2320	N/A
ipoaLisEntry	1.3.6.1.2.1.10.46.1.2.1	TCP/IP	RFC2320	N/A
ipoaLisSubnetAddr	1.3.6.1.2.1.10.46.1.2.1.1	TCP/IP	RFC2320	R/O
ipoaLisDefaultMtu	1.3.6.1.2.1.10.46.1.2.1.2	TCP/IP	RFC2320	R/O
ipoaLisDefaultEncapsType	1.3.6.1.2.1.10.46.1.2.1.3	TCP/IP	RFC2320	R/O
ipoaLisInactivityTimer	1.3.6.1.2.1.10.46.1.2.1.4	TCP/IP	RFC2320	R/O
ipoaLisMinHoldingTime	1.3.6.1.2.1.10.46.1.2.1.5	TCP/IP	RFC2320	R/O
ipoaLisQDepth	1.3.6.1.2.1.10.46.1.2.1.6	TCP/IP	RFC2320	R/O
ipoaLisMax Calls	1.3.6.1.2.1.10.46.1.2.1.7	TCP/IP	RFC2320	R/O
ipoaLisCacheEntryAge	1.3.6.1.2.1.10.46.1.2.1.8	TCP/IP	RFC2320	R/O
ipoaLisRetries	1.3.6.1.2.1.10.46.1.2.1.9	TCP/IP	RFC2320	R/O
ipoaLisTimeout	1.3.6.1.2.1.10.46.1.2.1.10	TCP/IP	RFC2320	R/O
ipoaLisDefaultPeakCellRate	1.3.6.1.2.1.10.46.1.2.1.11	TCP/IP	RFC2320	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ipoaLisActiveVcs	1.3.6.1.2.1.10.46.1.2.1.12	TCP/IP	RFC2320	R/O
ipoaLisIfMappingTable	1.3.6.1.2.1.10.46.1.3	TCP/IP	RFC2320	N/A
ipoaLisIfMappingEntry	1.3.6.1.2.1.10.46.1.3.1	TCP/IP	RFC2320	N/A
ipoaLisIfMappingRowStatus	1.3.6.1.2.1.10.46.1.3.1.1	TCP/IP	RFC2320	R/O
ipoaArpClientTable	1.3.6.1.2.1.10.46.1.4	TCP/IP	RFC2320	N/A
ipoaArpClientEntry	1.3.6.1.2.1.10.46.1.4.1	TCP/IP	RFC2320	N/A
ipoaArpClientAtmAddr	1.3.6.1.2.1.10.46.1.4.1.1	TCP/IP	RFC2320	R/O
ipoaArpClientSrvrInUse	1.3.6.1.2.1.10.46.1.4.1.2	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInReqs	1.3.6.1.2.1.10.46.1.4.1.3	TCP/IP	RFC2320	R/O
ipoaArpClientInArpOutReqs	1.3.6.1.2.1.10.46.1.4.1.4	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInReplies	1.3.6.1.2.1.10.46.1.4.1.5	TCP/IP	RFC2320	R/O
ipoaArpClientInArpOutReplies	1.3.6.1.2.1.10.46.1.4.1.6	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInvalidInReqs	1.3.6.1.2.1.10.46.1.4.1.7	TCP/IP	RFC2320	R/O
ipoaArpClientInArpInvalidOutReqs	1.3.6.1.2.1.10.46.1.4.1.8	TCP/IP	RFC2320	R/O
ipoaArpClientArpInReqs	1.3.6.1.2.1.10.46.1.4.1.9	TCP/IP	RFC2320	R/O
ipoaArpClientArpOutReqs	1.3.6.1.2.1.10.46.1.4.1.10	TCP/IP	RFC2320	R/O
ipoaArpClientArpInReplies	1.3.6.1.2.1.10.46.1.4.1.11	TCP/IP	RFC2320	R/O
ipoaArpClientArpOutReplies	1.3.6.1.2.1.10.46.1.4.1.12	TCP/IP	RFC2320	R/O
ipoaArpClientArpInNaks	1.3.6.1.2.1.10.46.1.4.1.13	TCP/IP	RFC2320	R/O
ipoaArpClientArpOutNaks	1.3.6.1.2.1.10.46.1.4.1.14	TCP/IP	RFC2320	R/O
ipoaArpClientArpUnknownOps	1.3.6.1.2.1.10.46.1.4.1.15	TCP/IP	RFC2320	R/O
ipoaArpClientArpNoSrvrResps	1.3.6.1.2.1.10.46.1.4.1.16	TCP/IP	RFC2320	R/O
ipoaArpRemoteSrvrTable	1.3.6.1.2.1.10.46.1.6	TCP/IP	RFC2320	N/A
ipoaArpRemoteSrvrEntry	1.3.6.1.2.1.10.46.1.6.1	TCP/IP	RFC2320	N/A
ipoaArpRemoteSrvrIpAddr	1.3.6.1.2.1.10.46.1.6.1.4	TCP/IP	RFC2320	R/O
ipoaVcTable	1.3.6.1.2.1.10.46.1.7	TCP/IP	RFC2320	N/A
ipoaVcEntry	1.3.6.1.2.1.10.46.1.7.1	TCP/IP	RFC2320	N/A
ipoaVcType	1.3.6.1.2.1.10.46.1.7.1.3	TCP/IP	RFC2320	R/O
ipoaVcNegotiatedEncapsType	1.3.6.1.2.1.10.46.1.7.1.4	TCP/IP	RFC2320	R/O
ipoaVcNegotiatedEncapsMtu	1.3.6.1.2.1.10.46.1.7.1.5	TCP/IP	RFC2320	R/O
ipoaConfigPvcTable	1.3.6.1.2.1.10.46.1.8	TCP/IP	RFC2320	N/A
ipoaConfigPvcEntry	1.3.6.1.2.1.10.46.1.8.1	TCP/IP	RFC2320	N/A
ipoaConfigPvcDefaultMtu	1.3.6.1.2.1.10.46.1.8.1.4	TCP/IP	RFC2320	R/O
atmInterfaceConfTable	1.3.6.1.2.1.37.1.2	TCP/IP	RFC1695	N/A

Object Descriptor	Object Identifier	Supported by	Defined by	AA
atmInterfaceConfEntry	1.3.6.1.2.1.37.1.2.1	TCP/IP	RFC1695	N/A
atmInterfaceMaxVpcs	1.3.6.1.2.1.37.1.2.1.1	TCP/IP	RFC1695	R/O
atmInterfaceMaxVccs	1.3.6.1.2.1.37.1.2.1.2	TCP/IP	RFC1695	R/O
atmInterfaceConfVpcs	1.3.6.1.2.1.37.1.2.1.3	TCP/IP	RFC1695	R/O
atmInterfaceConfVccs	1.3.6.1.2.1.37.1.2.1.4	TCP/IP	RFC1695	R/O
atmInterfaceMaxActiveVpiBits	1.3.6.1.2.1.37.1.2.1.5	TCP/IP	RFC1695	R/O
atmInterfaceMaxActiveVciBits	1.3.6.1.2.1.37.1.2.1.6	TCP/IP	RFC1695	R/O
atmInterfaceIlmiVpi	1.3.6.1.2.1.37.1.2.1.7	TCP/IP	RFC1695	R/O
atmInterfaceIlmiVci	1.3.6.1.2.1.37.1.2.1.8	TCP/IP	RFC1695	R/O
atmInterfaceAddressType	1.3.6.1.2.1.37.1.2.1.9	TCP/IP	RFC1695	R/O
atmInterfaceAdminAddress	1.3.6.1.2.1.37.1.2.1.10	TCP/IP	RFC1695	R/O
atmInterfaceMyNeighborIpAddress	1.3.6.1.2.1.37.1.2.1.11	TCP/IP	RFC1695	R/O
atmInterfaceMyNeighborIfName	1.3.6.1.2.1.37.1.2.1.12	TCP/IP	RFC1695	R/O
slapmSpinLock	1.3.6.1.3.88.1.1.1	SLA	SLAMIB	R/W
slapmPolicyPurgeTime	1.3.6.1.3.88.1.1.6	SLA	SLAMIB	R/W
slapmPolicyTrapEnable	1.3.6.1.3.88.1.1.7	SLA	SLAMIB	R/W
slapmPolicyTrapFilter	1.3.6.1.3.88.1.1.8	SLA	SLAMIB	R/W
slapmPolicyStatsTable	1.3.6.1.3.88.1.2.1	SLA	SLAMIB	N/A
slapmPolicyStatsEntry	1.3.6.1.3.88.1.2.1.1	SLA	SLAMIB	N/A
slapmPolicyStatsSystemAddress	1.3.6.1.3.88.1.2.1.1.1	SLA	SLAMIB	N/A
slapmPolicyStatsPolicyName	1.3.6.1.3.88.1.2.1.1.2	SLA	SLAMIB	N/A
slapmPolicyStatsTrafficProfileName	1.3.6.1.3.88.1.2.1.1.3	SLA	SLAMIB	N/A
slapmPolicyStatsOperStatus	1.3.6.1.3.88.1.2.1.1.4	SLA	SLAMIB	R/O
slapmPolicyStatsActiveConns	1.3.6.1.3.88.1.2.1.1.5	SLA	SLAMIB	R/O
slapmPolicyStatsTotalConns	1.3.6.1.3.88.1.2.1.1.6	SLA	SLAMIB	R/O
slapmPolicyStatsFirstActivated	1.3.6.1.3.88.1.2.1.1.7	SLA	SLAMIB	R/O
slapmPolicyStatsLastMapping	1.3.6.1.3.88.1.2.1.1.8	SLA	SLAMIB	R/O
slapmPolicyStatsInOctets	1.3.6.1.3.88.1.2.1.1.9	SLA	SLAMIB	R/O
slapmPolicyStatsOutOctets	1.3.6.1.3.88.1.2.1.1.10	SLA	SLAMIB	R/O
slapmPolicyStatsConnectionLimit	1.3.6.1.3.88.1.2.1.1.11	SLA	SLAMIB	R/O
slapmPolicyStatsCountAccepts	1.3.6.1.3.88.1.2.1.1.12	SLA	SLAMIB	R/O
slapmPolicyStatsCountDenies	1.3.6.1.3.88.1.2.1.1.13	SLA	SLAMIB	R/O
slapmPolicyStatsInDiscards	1.3.6.1.3.88.1.2.1.1.14	SLA	SLAMIB	R/O
slapmPolicyStatsOutDiscards	1.3.6.1.3.88.1.2.1.1.15	SLA	SLAMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
slapmPolicyStatsInPackets	1.3.6.1.3.88.1.2.1.1.16	SLA	SLAMIB	R/O
slapmPolicyStatsOutPackets	1.3.6.1.3.88.1.2.1.1.17	SLA	SLAMIB	R/O
slapmPolicyStatsMinRate	1.3.6.1.3.88.1.2.1.1.20	SLA	SLAMIB	R/O
slapmPolicyStatsMaxRate	1.3.6.1.3.88.1.2.1.1.21	SLA	SLAMIB	R/O
slapmPolicyStatsMaxDelay	1.3.6.1.3.88.1.2.1.1.22	SLA	SLAMIB	R/O
slapmPolicyMonitorTable	1.3.6.1.3.88.1.2.2	SLA	SLAMIB	N/A
slapmPolicyMonitorEntry	1.3.6.1.3.88.1.2.2.1	SLA	SLAMIB	N/A
slapmPolicyMonitorOwnerIndex	1.3.6.1.3.88.1.2.2.1.1	SLA	SLAMIB	N/A
slapmPolicyMonitorSystemAddress	1.3.6.1.3.88.1.2.2.1.2	SLA	SLAMIB	N/A
slapmPolicyMonitorPolicyName	1.3.6.1.3.88.1.2.2.1.3	SLA	SLAMIB	N/A
slapmPolicyMonitorTrafficProfileName	1.3.6.1.3.88.1.2.2.1.4	SLA	SLAMIB	N/A
slapmPolicyMonitorControl	1.3.6.1.3.88.1.2.2.1.5	SLA	SLAMIB	R/C
slapmPolicyMonitorStatus	1.3.6.1.3.88.1.2.2.1.6	SLA	SLAMIB	R/O
slapmPolicyMonitorInterval	1.3.6.1.3.88.1.2.2.1.7	SLA	SLAMIB	R/C
slapmPolicyMonitorIntTime	1.3.6.1.3.88.1.2.2.1.8	SLA	SLAMIB	R/O
slapmPolicyMonitorCurrentInRate	1.3.6.1.3.88.1.2.2.1.9	SLA	SLAMIB	R/O
slapmPolicyMonitorCurrentOutRate	1.3.6.1.3.88.1.2.2.1.10	SLA	SLAMIB	R/O
slapmPolicyMonitorMinRateLow	1.3.6.1.3.88.1.2.2.1.11	SLA	SLAMIB	R/C
slapmPolicyMonitorMinRateHigh	1.3.6.1.3.88.1.2.2.1.12	SLA	SLAMIB	R/C
slapmPolicyMonitorMaxRateHigh	1.3.6.1.3.88.1.2.2.1.13	SLA	SLAMIB	R/C
slapmPolicyMonitorMaxRateLow	1.3.6.1.3.88.1.2.2.1.14	SLA	SLAMIB	R/C
slapmPolicyMonitorMaxDelayHigh	1.3.6.1.3.88.1.2.2.1.15	SLA	SLAMIB	R/C
slapmPolicyMonitorMaxDelayLow	1.3.6.1.3.88.1.2.2.1.16	SLA	SLAMIB	R/C
slapmPolicyMonitorMinInRateNotAchieves	1.3.6.1.3.88.1.2.2.1.17	SLA	SLAMIB	R/O
slapmPolicyMonitorMaxInRateExceeds	1.3.6.1.3.88.1.2.2.1.18	SLA	SLAMIB	R/O
slapmPolicyMonitorMaxDelayExceeds	1.3.6.1.3.88.1.2.2.1.19	SLA	SLAMIB	R/O
slapmPolicyMonitorMinOutRateNotAchieves	1.3.6.1.3.88.1.2.2.1.20	SLA	SLAMIB	R/O
slapmPolicyMonitorMaxOutRateExceeds	1.3.6.1.3.88.1.2.2.1.21	SLA	SLAMIB	R/O
slapmPolicyMonitorCurrentDelayRate	1.3.6.1.3.88.1.2.2.1.22	SLA	SLAMIB	R/O
slapmPolicyMonitorRowStatus	1.3.6.1.3.88.1.2.2.1.23	SLA	SLAMIB	R/C
slapmSubcomponentTable	1.3.6.1.3.88.1.2.3	SLA	SLAMIB	N/A
slapmSubcomponentEntry	1.3.6.1.3.88.1.2.3.1	SLA	SLAMIB	R/O
slapmSubcomponentRemAddress	1.3.6.1.3.88.1.2.3.1.1	SLA	SLAMIB	N/A
slapmSubcomponentRemPort	1.3.6.1.3.88.1.2.3.1.2	SLA	SLAMIB	N/A

Object Descriptor	Object Identifier	Supported by	Defined by	AA
slapmSubcomponentLocalAddress	1.3.6.1.3.88.1.2.3.1.3	SLA	SLAMIB	N/A
slapmSubcomponentLocalPort	1.3.6.1.3.88.1.2.3.1.4	SLA	SLAMIB	N/A
slapmSubcomponentProtocol	1.3.6.1.3.88.1.2.3.1.5	SLA	SLAMIB	R/O
slapmSubcomponentSystemAddress	1.3.6.1.3.88.1.2.3.1.6	SLA	SLAMIB	R/O
slapmSubcomponentPolicyName	1.3.6.1.3.88.1.2.3.1.7	SLA	SLAMIB	R/O
slapmSubcomponentTrafficProfileName	1.3.6.1.3.88.1.2.3.1.8	SLA	SLAMIB	R/O
slapmSubcomponentLastActivity	1.3.6.1.3.88.1.2.3.1.9	SLA	SLAMIB	R/O
slapmSubcomponentInOctets	1.3.6.1.3.88.1.2.3.1.10	SLA	SLAMIB	R/O
slapmSubcomponentOutOctets	1.3.6.1.3.88.1.2.3.1.11	SLA	SLAMIB	R/O
slapmSubcomponentTcpOutBufferedOctets	1.3.6.1.3.88.1.2.3.1.12	SLA	SLAMIB	R/O
slapmSubcomponentTcpInBufferedOctets	1.3.6.1.3.88.1.2.3.1.13	SLA	SLAMIB	R/O
slapmSubcomponentTcpReXmts	1.3.6.1.3.88.1.2.3.1.14	SLA	SLAMIB	R/O
slapmSubcomponentTcpRoundTripTime	1.3.6.1.3.88.1.2.3.1.15	SLA	SLAMIB	R/O
slapmSubcomponentTcpRoundTripVariance	1.3.6.1.3.88.1.2.3.1.16	SLA	SLAMIB	R/O
slapmSubcomponentInPdus	1.3.6.1.3.88.1.2.3.1.17	SLA	SLAMIB	R/O
slapmSubcomponentOutPdus	1.3.6.1.3.88.1.2.3.1.18	SLA	SLAMIB	R/O
slapmSubcomponentApplName	1.3.6.1.3.88.1.2.3.1.19	SLA	SLAMIB	R/O
slapmSubcomponentMonitorStatus	1.3.6.1.3.88.1.2.3.1.20	SLA	SLAMIB	R/O
slapmSubcomponentMonitorIntTime	1.3.6.1.3.88.1.2.3.1.21	SLA	SLAMIB	R/O
slapmSubcomponentMonitorCurrentInRate	1.3.6.1.3.88.1.2.3.1.22	SLA	SLAMIB	R/O
slapmSubcomponentMonitorCurrentOutRate	1.3.6.1.3.88.1.2.3.1.23	SLA	SLAMIB	R/O
dpiPort	1.3.6.1.4.1.2.2.1.1.0	Agent	RFC1592	R/O
dpiPortForTCP	1.3.6.1.4.1.2.2.1.1.1.0	Agent	RFC1592	R/O
dpiPortForUDP	1.3.6.1.4.1.2.2.1.1.2.0	Agent	RFC1592	R/O
dpiPathNameForUnixStream	1.3.6.1.4.1.2.2.1.1.3.0	Agent	RFC1592	R/O
saDefaultTimeout	1.3.6.1.4.1.2.4.12.1	Agent	SAMIB	R/W
saMaxTimeout	1.3.6.1.4.1.2.4.12.2	Agent	SAMIB	R/W
saAllowDuplicateIDs	1.3.6.1.4.1.2.4.12.3	Agent	SAMIB	R/W
saNumber	1.3.6.1.4.1.2.4.12.4	Agent	SAMIB	R/O
saAllPacketsIn	1.3.6.1.4.1.2.4.12.5	Agent	SAMIB	R/O
saAllPacketsOut	1.3.6.1.4.1.2.4.12.6	Agent	SAMIB	R/O
saTable	1.3.6.1.4.1.2.4.12.7	Agent	SAMIB	N/A
saEntry	1.3.6.1.4.1.2.4.12.7.1	Agent	SAMIB	N/A
saIndex	1.3.6.1.4.1.2.4.12.7.1.1	Agent	SAMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
saIdentifier	1.3.6.1.4.1.2.4.12.7.1.2	Agent	SAMIB	R/O
saDescription	1.3.6.1.4.1.2.4.12.7.1.3	Agent	SAMIB	R/O
saStatus	1.3.6.1.4.1.2.4.12.7.1.4	Agent	SAMIB	R/W
saStatusChangeTime	1.3.6.1.4.1.2.4.12.7.1.5	Agent	SAMIB	R/O
saProtocol	1.3.6.1.4.1.2.4.12.7.1.6	Agent	SAMIB	R/O
saProtocolVersion	1.3.6.1.4.1.2.4.12.7.1.7	Agent	SAMIB	R/O
saProtocolRelease	1.3.6.1.4.1.2.4.12.7.1.8	Agent	SAMIB	R/O
saTransport	1.3.6.1.4.1.2.4.12.7.1.9	Agent	SAMIB	R/O
saTransportAddress	1.3.6.1.4.1.2.4.12.7.1.10	Agent	SAMIB	R/O
saTimeout	1.3.6.1.4.1.2.4.12.7.1.11	Agent	SAMIB	R/W
saMaxVarBinds	1.3.6.1.4.1.2.4.12.7.1.12	Agent	SAMIB	R/O
saPacketsIn	1.3.6.1.4.1.2.4.12.7.1.13	Agent	SAMIB	R/O
saPacketsOut	1.3.6.1.4.1.2.4.12.7.1.14	Agent	SAMIB	R/O
saTreeTable	1.3.6.1.4.1.2.4.12.8	Agent	SAMIB	N/A
saTreeEntry	1.3.6.1.4.1.2.4.12.8.1	Agent	SAMIB	N/A
saTsubtree	1.3.6.1.4.1.2.4.12.8.1.1	Agent	SAMIB	R/O
saTpriority	1.3.6.1.4.1.2.4.12.8.1.2	Agent	SAMIB	R/O
saTindex	1.3.6.1.4.1.2.4.12.8.1.3	Agent	SAMIB	R/O
saTstatus	1.3.6.1.4.1.2.4.12.8.1.4	Agent	SAMIB	R/W
saTtimeout	1.3.6.1.4.1.2.4.12.8.1.5	Agent	SAMIB	R/W
ibm3172Descr	1.3.6.1.4.1.2.6.1.1.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172Contact	1.3.6.1.4.1.2.6.1.1.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172Location	1.3.6.1.4.1.2.6.1.1.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifNumber	1.3.6.1.4.1.2.6.1.1.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifTrapEnable	1.3.6.1.4.1.2.6.1.2.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifInChanOctets	1.3.6.1.4.1.2.6.1.3.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutChanOctets	1.3.6.1.4.1.2.6.1.3.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifInChanBlocks	1.3.6.1.4.1.2.6.1.3.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutChanBlocks	1.3.6.1.4.1.2.6.1.3.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANOctets	1.3.6.1.4.1.2.6.1.4.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANOctets	1.3.6.1.4.1.2.6.1.4.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANFrames	1.3.6.1.4.1.2.6.1.4.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANFrames	1.3.6.1.4.1.2.6.1.4.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANErrors	1.3.6.1.4.1.2.6.1.4.1.5	TCP/IP	ibm3172MIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibm3172ifOutLANErrors	1.3.6.1.4.1.2.6.1.4.1.6	TCP/IP	ibm3172MIB	R/O
ibm3172ifInLANDiscards	1.3.6.1.4.1.2.6.1.4.1.7	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutLANDiscards	1.3.6.1.4.1.2.6.1.4.1.8	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkRcvOctets	1.3.6.1.4.1.2.6.1.5.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkXmitOctets	1.3.6.1.4.1.2.6.1.5.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkRcvFrames	1.3.6.1.4.1.2.6.1.5.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifBlkXmitBlocks	1.3.6.1.4.1.2.6.1.5.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifInBlkErrors	1.3.6.1.4.1.2.6.1.5.1.5	TCP/IP	ibm3172MIB	R/O
ibm3172ifInBlkDiscards	1.3.6.1.4.1.2.6.1.5.1.6	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkRcvOctets	1.3.6.1.4.1.2.6.1.6.1.1	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkXmitOctets	1.3.6.1.4.1.2.6.1.6.1.2	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkRcvBlocks	1.3.6.1.4.1.2.6.1.6.1.3	TCP/IP	ibm3172MIB	R/O
ibm3172ifDblkXmitFrames	1.3.6.1.4.1.2.6.1.6.1.4	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutDblkErrors	1.3.6.1.4.1.2.6.1.6.1.5	TCP/IP	ibm3172MIB	R/O
ibm3172ifOutDblkDiscards	1.3.6.1.4.1.2.6.1.6.1.6	TCP/IP	ibm3172MIB	R/O
ibm3172ifDeviceNumber	1.3.6.1.4.1.2.6.1.7.1.1	TCP/IP	ibm3172MIB	R/O
ibmRemotePingTable	1.3.6.1.4.1.2.6.19.2.2.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmRemotePingEntry	1.3.6.1.4.1.2.6.19.2.2.1.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsRPingResponseTime	1.3.6.1.4.1.2.6.19.2.2.1.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSubagentCacheTime	1.3.6.1.4.1.2.6.19.2.2.2.1	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIgnoreRedirect	1.3.6.1.4.1.2.6.19.2.2.2.2	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsArpCacheTimeout	1.3.6.1.4.1.2.6.19.2.2.2.3	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpKeepAliveTimer	1.3.6.1.4.1.2.6.19.2.2.2.4	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpReceiveBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.5	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpSendBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.6	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpChecksum	1.3.6.1.4.1.2.6.19.2.2.2.7	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIplDateAndTime	1.3.6.1.4.1.2.6.19.2.2.2.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsNoUdpQueueLimit	1.3.6.1.4.1.2.6.19.2.2.2.9	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsSoMaxConn	1.3.6.1.4.1.2.6.19.2.2.2.10	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpipProcname	1.3.6.1.4.1.2.6.19.2.2.2.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpipAsid	1.3.6.1.4.1.2.6.19.2.2.2.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSourceVipaEnabled	1.3.6.1.4.1.2.6.19.2.2.2.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsasfSysplexName	1.3.6.1.4.1.2.6.19.2.2.2.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsOsasfHostName	1.3.6.1.4.1.2.6.19.2.2.2.15	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsOsasfProductVersion	1.3.6.1.4.1.2.6.19.2.2.2.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPrimaryInterfaceIfIndex	1.3.6.1.4.1.2.6.19.2.2.2.17	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIpMaxReassemblySize	1.3.6.1.4.1.2.6.19.2.2.2.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpRestrictLowPorts	1.3.6.1.4.1.2.6.19.2.2.2.19	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpRestrictLowPorts	1.3.6.1.4.1.2.6.19.2.2.2.20	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpSendBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.21	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpRecvBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.22	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsTcpipStatisticsEnabled	1.3.6.1.4.1.2.6.19.2.2.2.23	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsFirewallEnabled	1.3.6.1.4.1.2.6.19.2.2.2.24	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsMaximumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.2.25	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsMinimumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.2.26	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsRoundTripGain	1.3.6.1.4.1.2.6.19.2.2.2.27	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsVarianceGain	1.3.6.1.4.1.2.6.19.2.2.2.28	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsVarianceMultiplier	1.3.6.1.4.1.2.6.19.2.2.2.29	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsSendGarbageEnabled	1.3.6.1.4.1.2.6.19.2.2.2.30	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpMaxReceiveBufferSize	1.3.6.1.4.1.2.6.19.2.2.2.31	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsPathMtuDscEnabled	1.3.6.1.4.1.2.6.19.2.2.2.33	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsMultipathType	1.3.6.1.4.1.2.6.19.2.2.2.34	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsIpForwarding	1.3.6.1.4.1.2.6.19.2.2.2.35	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsDeviceType	1.3.6.1.4.1.2.6.19.2.2.3.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceBaseNumber	1.3.6.1.4.1.2.6.19.2.2.3.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceIoBufferSize	1.3.6.1.4.1.2.6.19.2.2.3.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceAutoRestart	1.3.6.1.4.1.2.6.19.2.2.3.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceNetmanEnabled	1.3.6.1.4.1.2.6.19.2.2.3.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceHostClawName	1.3.6.1.4.1.2.6.19.2.2.3.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceWorkstationClawName	1.3.6.1.4.1.2.6.19.2.2.3.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceReadBuffers	1.3.6.1.4.1.2.6.19.2.2.3.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceReadSize	1.3.6.1.4.1.2.6.19.2.2.3.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceWriteBuffers	1.3.6.1.4.1.2.6.19.2.2.3.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceWriteSize	1.3.6.1.4.1.2.6.19.2.2.3.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceProcname	1.3.6.1.4.1.2.6.19.2.2.3.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceIncomingSvcEnabled	1.3.6.1.4.1.2.6.19.2.2.3.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceLuName	1.3.6.1.4.1.2.6.19.2.2.3.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsDeviceRouterStatus	1.3.6.1.4.1.2.6.19.2.2.3.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsLinkType	1.3.6.1.4.1.2.6.19.2.2.3.2.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkDeviceIndex	1.3.6.1.4.1.2.6.19.2.2.3.2.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkAdapterAddr	1.3.6.1.4.1.2.6.19.2.2.3.2.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkNumber	1.3.6.1.4.1.2.6.19.2.2.3.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkIbmtrCanonical	1.3.6.1.4.1.2.6.19.2.2.3.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkIbmtrBcast	1.3.6.1.4.1.2.6.19.2.2.3.2.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkMcast	1.3.6.1.4.1.2.6.19.2.2.3.2.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkChecksumEnabled	1.3.6.1.4.1.2.6.19.2.2.3.2.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsLinkMcastRefCount	1.3.6.1.4.1.2.6.19.2.2.3.3.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortNumberLow	1.3.6.1.4.1.2.6.19.2.2.4.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortNumberHigh	1.3.6.1.4.1.2.6.19.2.2.4.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortProtocol	1.3.6.1.4.1.2.6.19.2.2.4.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortProcName	1.3.6.1.4.1.2.6.19.2.2.4.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortAutoLoggable	1.3.6.1.4.1.2.6.19.2.2.4.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortDelayAcks	1.3.6.1.4.1.2.6.19.2.2.4.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortOptMaxSegmentSize	1.3.6.1.4.1.2.6.19.2.2.4.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsPortSharePort	1.3.6.1.4.1.2.6.19.2.2.4.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayMaximumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.5.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayMinimumRetransmitTime	1.3.6.1.4.1.2.6.19.2.2.5.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayRoundTripGain	1.3.6.1.4.1.2.6.19.2.2.5.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayVarianceGain	1.3.6.1.4.1.2.6.19.2.2.5.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayVarianceMultiplier	1.3.6.1.4.1.2.6.19.2.2.5.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsGatewayDelayAcks	1.3.6.1.4.1.2.6.19.2.2.5.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
osasfChannelTable	1.3.6.1.4.1.2.6.19.2.2.6.1	TCP/IP	ibmTCPIPmvsMIB	N/A
osasfChannelEntry	1.3.6.1.4.1.2.6.19.2.2.6.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfChannelNumber	1.3.6.1.4.1.2.6.19.2.2.6.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelType	1.3.6.1.4.1.2.6.19.2.2.6.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelSubType	1.3.6.1.4.1.2.6.19.2.2.6.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelMode	1.3.6.1.4.1.2.6.19.2.2.6.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelHwModel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelState	1.3.6.1.4.1.2.6.19.2.2.6.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelShared	1.3.6.1.4.1.2.6.19.2.2.6.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelNumPorts	1.3.6.1.4.1.2.6.19.2.2.6.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelDeterNodeDesc	1.3.6.1.4.1.2.6.19.2.2.6.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsAtmOsasfChannelControlUnitNumber	1.3.6.1.4.1.2.6.19.2.2.6.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCodeLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelEcLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCurLparName	1.3.6.1.4.1.2.6.19.2.2.6.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelCurLparNum	1.3.6.1.4.1.2.6.19.2.2.6.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelManParnName	1.3.6.1.4.1.2.6.19.2.2.6.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelManParnNum	1.3.6.1.4.1.2.6.19.2.2.6.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelDate	1.3.6.1.4.1.2.6.19.2.2.6.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelTime	1.3.6.1.4.1.2.6.19.2.2.6.1.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfChannelFlashLevel	1.3.6.1.4.1.2.6.19.2.2.6.1.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
osasfPortTable	1.3.6.1.4.1.2.6.19.2.2.6.2	TCP/IP	ibmTCPIPmvsMIB	N/A
osasfPortEntry	1.3.6.1.4.1.2.6.19.2.2.6.2.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfPortNumber	1.3.6.1.4.1.2.6.19.2.2.6.2.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortHardwareState	1.3.6.1.4.1.2.6.19.2.2.6.2.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMediaType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortUniType	1.3.6.1.4.1.2.6.19.2.2.6.2.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortUniVersion	1.3.6.1.4.1.2.6.19.2.2.6.2.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefix	1.3.6.1.4.1.2.6.19.2.2.6.2.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefixPrefix	1.3.6.1.4.1.2.6.19.2.2.6.2.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortNetPrefixStatus	1.3.6.1.4.1.2.6.19.2.2.6.2.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortCodeLoadStatus	1.3.6.1.4.1.2.6.19.2.2.6.2.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMacAddrBurntIn	1.3.6.1.4.1.2.6.19.2.2.6.2.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMacAddrActive	1.3.6.1.4.1.2.6.19.2.2.6.2.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortMaxPcmConnections	1.3.6.1.4.1.2.6.19.2.2.6.2.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortPcmName	1.3.6.1.4.1.2.6.19.2.2.6.2.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortAAL5InPackets	1.3.6.1.4.1.2.6.19.2.2.6.2.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortAAL5OutPackets	1.3.6.1.4.1.2.6.19.2.2.6.2.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPortIpAddress	1.3.6.1.4.1.2.6.19.2.2.6.2.1.17	TCP/IP	ibmTCPIPmvsMIB	R/W
osasfPvcTable	1.3.6.1.4.1.2.6.19.2.2.6.3	TCP/IP	ibmTCPIPmvsMIB	N/A
osasfPvcEntry	1.3.6.1.4.1.2.6.19.2.2.6.3.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmOsasfPvcName	1.3.6.1.4.1.2.6.19.2.2.6.3.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBestEffort	1.3.6.1.4.1.2.6.19.2.2.6.3.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdPeakCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsAtmOsasfPvcBwdPeakCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdsustainCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdsustainCellRate	1.3.6.1.4.1.2.6.19.2.2.6.3.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdCellBurstSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdCellBurstSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcVpi	1.3.6.1.4.1.2.6.19.2.2.6.3.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcVci	1.3.6.1.4.1.2.6.19.2.2.6.3.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcFwdMaxAal5PduSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmOsasfPvcBwdMaxAal5PduSize	1.3.6.1.4.1.2.6.19.2.2.6.3.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTable	1.3.6.1.4.1.2.6.19.2.2.6.4	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmSnaLeEntry	1.3.6.1.4.1.2.6.19.2.2.6.4.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmSnaLeLlcTi	1.3.6.1.4.1.2.6.19.2.2.6.4.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeLlcT1	1.3.6.1.4.1.2.6.19.2.2.6.4.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeLlcT2	1.3.6.1.4.1.2.6.19.2.2.6.4.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxStations	1.3.6.1.4.1.2.6.19.2.2.6.4.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxSaps	1.3.6.1.4.1.2.6.19.2.2.6.4.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxIn	1.3.6.1.4.1.2.6.19.2.2.6.4.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxOut	1.3.6.1.4.1.2.6.19.2.2.6.4.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeCrsGroupAddress	1.3.6.1.4.1.2.6.19.2.2.6.4.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeCrsUserData	1.3.6.1.4.1.2.6.19.2.2.6.4.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeClientEnableState	1.3.6.1.4.1.2.6.19.2.2.6.4.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeBestEffortPeakRate	1.3.6.1.4.1.2.6.19.2.2.6.4.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeMaxLECConnections	1.3.6.1.4.1.2.6.19.2.2.6.4.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTrEnableLoadBalancing	1.3.6.1.4.1.2.6.19.2.2.6.4.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTrLoadBalancing	1.3.6.1.4.1.2.6.19.2.2.6.4.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmSnaLeTrSessionDelay	1.3.6.1.4.1.2.6.19.2.2.6.4.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigTable	1.3.6.1.4.1.2.6.19.2.2.6.5	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecConfigEntry	1.3.6.1.4.1.2.6.19.2.2.6.5.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecConfigMode	1.3.6.1.4.1.2.6.19.2.2.6.5.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigLanType	1.3.6.1.4.1.2.6.19.2.2.6.5.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMaxDataFrameSize	1.3.6.1.4.1.2.6.19.2.2.6.5.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigLanName	1.3.6.1.4.1.2.6.19.2.2.6.5.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigLesAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.5.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlTimeout	1.3.6.1.4.1.2.6.19.2.2.6.5.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsAtmLecMaxUnknownFrameCount	1.3.6.1.4.1.2.6.19.2.2.6.5.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecUnknownFrameTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecVccTimeoutPeriod	1.3.6.1.4.1.2.6.19.2.2.6.5.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMaxRetryCount	1.3.6.1.4.1.2.6.19.2.2.6.5.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecAgingTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecForwardDelayTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecExpectedArpResponseTime	1.3.6.1.4.1.2.6.19.2.2.6.5.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecFlushTimeout	1.3.6.1.4.1.2.6.19.2.2.6.5.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecPathSwitchingDelay	1.3.6.1.4.1.2.6.19.2.2.6.5.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecLocalSegmentID	1.3.6.1.4.1.2.6.19.2.2.6.5.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendType	1.3.6.1.4.1.2.6.19.2.2.6.5.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendAvgRate	1.3.6.1.4.1.2.6.19.2.2.6.5.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendPeakRate	1.3.6.1.4.1.2.6.19.2.2.6.5.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConnectionCompleteTimer	1.3.6.1.4.1.2.6.19.2.2.6.5.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecStatusTable	1.3.6.1.4.1.2.6.19.2.2.6.6	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecStatusEntry	1.3.6.1.4.1.2.6.19.2.2.6.6.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecPrimaryAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.6.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecID	1.3.6.1.4.1.2.6.19.2.2.6.6.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecInterfaceState	1.3.6.1.4.1.2.6.19.2.2.6.6.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecLastFailureRespCode	1.3.6.1.4.1.2.6.19.2.2.6.6.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecLastFailureState	1.3.6.1.4.1.2.6.19.2.2.6.6.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecProtocol	1.3.6.1.4.1.2.6.19.2.2.6.6.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecVersion	1.3.6.1.4.1.2.6.19.2.2.6.6.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecTopologyChange	1.3.6.1.4.1.2.6.19.2.2.6.6.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecconfigServerAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.6.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigSource	1.3.6.1.4.1.2.6.19.2.2.6.6.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecActualLanType	1.3.6.1.4.1.2.6.19.2.2.6.6.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecActualMaxDataFrameSize	1.3.6.1.4.1.2.6.19.2.2.6.6.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecActualLanName	1.3.6.1.4.1.2.6.19.2.2.6.6.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecAtmAddress	1.3.6.1.4.1.2.6.19.2.2.6.6.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecProxyClient	1.3.6.1.4.1.2.6.19.2.2.6.6.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecStatisticsTable	1.3.6.1.4.1.2.6.19.2.2.6.7	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecStatisticsEntry	1.3.6.1.4.1.2.6.19.2.2.6.7.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecArpRequestsOut	1.3.6.1.4.1.2.6.19.2.2.6.7.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsAtmLecArpRequestsIn	1.3.6.1.4.1.2.6.19.2.2.6.7.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecArpRepliesOut	1.3.6.1.4.1.2.6.19.2.2.6.7.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecArpRepliesIn	1.3.6.1.4.1.2.6.19.2.2.6.7.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlFramesOut	1.3.6.1.4.1.2.6.19.2.2.6.7.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlFramesIn	1.3.6.1.4.1.2.6.19.2.2.6.7.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecSvcFailures	1.3.6.1.4.1.2.6.19.2.2.6.7.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecServerTable	1.3.6.1.4.1.2.6.19.2.2.6.8	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecServerEntry	1.3.6.1.4.1.2.6.19.2.2.6.8.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecConfigDirectInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigDirectVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecConfigDirectVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDirectInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDirectVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDirectVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDistributeInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDistributeVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecControlDistributeVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastSendVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastFwdInterface	1.3.6.1.4.1.2.6.19.2.2.6.8.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastFwdVPI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.14	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMulticastFwdVCI	1.3.6.1.4.1.2.6.19.2.2.6.8.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsAtmLecMacAddressTable	1.3.6.1.4.1.2.6.19.2.2.6.9	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecMacAddressEntry	1.3.6.1.4.1.2.6.19.2.2.6.9.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsAtmLecMacAddress	1.3.6.1.4.1.2.6.19.2.2.6.9.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsTcpConnTable	1.3.6.1.4.1.2.6.19.2.2.7.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsTcpConnEntry	1.3.6.1.4.1.2.6.19.2.2.7.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsTcpConnLastActivity	1.3.6.1.4.1.2.6.19.2.2.7.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnBytesIn	1.3.6.1.4.1.2.6.19.2.2.7.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnBytesOut	1.3.6.1.4.1.2.6.19.2.2.7.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIpTos	1.3.6.1.4.1.2.6.19.2.2.7.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOptions	1.3.6.1.4.1.2.6.19.2.2.7.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutBuffered	1.3.6.1.4.1.2.6.19.2.2.7.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsTcpConnUsrSndNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndUna	1.3.6.1.4.1.2.6.19.2.2.7.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingPush	1.3.6.1.4.1.2.6.19.2.2.7.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingUrg	1.3.6.1.4.1.2.6.19.2.2.7.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutgoingWinSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInBuffered	1.3.6.1.4.1.2.6.19.2.2.7.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRcvNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnUsrRcvNxt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.18	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingPush	1.3.6.1.4.1.2.6.19.2.2.7.1.1.19	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingUrg	1.3.6.1.4.1.2.6.19.2.2.7.1.1.20	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnIncomingWinSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.21	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnReXmt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.22	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnMaxSndWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.23	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnReXmtCount	1.3.6.1.4.1.2.6.19.2.2.7.1.1.24	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnCongestionWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.25	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSSThresh	1.3.6.1.4.1.2.6.19.2.2.7.1.1.26	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRoundTripTime	1.3.6.1.4.1.2.6.19.2.2.7.1.1.27	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRoundTripVariance	1.3.6.1.4.1.2.6.19.2.2.7.1.1.28	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInitSndSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.29	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInitRcvSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.30	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSendMSS	1.3.6.1.4.1.2.6.19.2.2.7.1.1.31	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWl1	1.3.6.1.4.1.2.6.19.2.2.7.1.1.32	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWl2	1.3.6.1.4.1.2.6.19.2.2.7.1.1.33	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSndWnd	1.3.6.1.4.1.2.6.19.2.2.7.1.1.34	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRcvBufSize	1.3.6.1.4.1.2.6.19.2.2.7.1.1.36	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnResourceName	1.3.6.1.4.1.2.6.19.2.2.7.1.1.37	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSubtask	1.3.6.1.4.1.2.6.19.2.2.7.1.1.38	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnResourceId	1.3.6.1.4.1.2.6.19.2.2.7.1.1.39	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnSockOpt	1.3.6.1.4.1.2.6.19.2.2.7.1.1.40	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpTimer	1.3.6.1.4.1.2.6.19.2.2.7.1.1.41	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpSig	1.3.6.1.4.1.2.6.19.2.2.7.1.1.42	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpSel	1.3.6.1.4.1.2.6.19.2.2.7.1.1.43	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnRttSeq	1.3.6.1.4.1.2.6.19.2.2.7.1.1.44	TCP/IP	ibmTCPIPmvsMIB	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
ibmMvsTcpConnTcpDet	1.3.6.1.4.1.2.6.19.2.2.7.1.1.46	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTcpPol	1.3.6.1.4.1.2.6.19.2.2.7.1.1.47	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnTargetAppl	1.3.6.1.4.1.2.6.19.2.2.7.1.1.48	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnLuName	1.3.6.1.4.1.2.6.19.2.2.7.1.1.49	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnClientUserID	1.3.6.1.4.1.2.6.19.2.2.7.1.1.50	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnLogMode	1.3.6.1.4.1.2.6.19.2.2.7.1.1.51	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnProto	1.3.6.1.4.1.2.6.19.2.2.7.1.1.52	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnDupacks	1.3.6.1.4.1.2.6.19.2.2.7.1.1.53	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOptMaxSegmentSize	1.3.6.1.4.1.2.6.19.2.2.7.1.1.54	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnClusterConnFlag	1.3.6.1.4.1.2.6.19.2.2.7.1.1.55	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnInSegs	1.3.6.1.4.1.2.6.19.2.2.7.1.1.56	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsTcpConnOutSegs	1.3.6.1.4.1.2.6.19.2.2.7.1.1.57	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmTcpipMvsUdpTable	1.3.6.1.4.1.2.6.19.2.2.8.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmTcpipMvsUdpEntry	1.3.6.1.4.1.2.6.19.2.2.8.1.1	TCP/IP	ibmTCPIPmvsMIB	N/A
ibmMvsUdpLastAct	1.3.6.1.4.1.2.6.19.2.2.8.1.1.1	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpTos	1.3.6.1.4.1.2.6.19.2.2.8.1.1.2	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpIpOpts	1.3.6.1.4.1.2.6.19.2.2.8.1.1.3	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDgramIn	1.3.6.1.4.1.2.6.19.2.2.8.1.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpBytesIn	1.3.6.1.4.1.2.6.19.2.2.8.1.1.5	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpDgramOut	1.3.6.1.4.1.2.6.19.2.2.8.1.1.6	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpBytesOut	1.3.6.1.4.1.2.6.19.2.2.8.1.1.7	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpResourceName	1.3.6.1.4.1.2.6.19.2.2.8.1.1.8	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSubtask	1.3.6.1.4.1.2.6.19.2.2.8.1.1.9	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpResourceId	1.3.6.1.4.1.2.6.19.2.2.8.1.1.10	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSockOpt	1.3.6.1.4.1.2.6.19.2.2.8.1.1.11	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpSendLim	1.3.6.1.4.1.2.6.19.2.2.8.1.1.12	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpRecvLim	1.3.6.1.4.1.2.6.19.2.2.8.1.1.13	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpEntryState	1.3.6.1.4.1.2.6.19.2.2.8.1.1.14	TCP/IP	ibmTCPIPmvsMIB	R/W
ibmMvsUdpMcastTTL	1.3.6.1.4.1.2.6.19.2.2.8.1.1.15	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpMcastLoopback	1.3.6.1.4.1.2.6.19.2.2.8.1.1.16	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpMcastLinkAddr	1.3.6.1.4.1.2.6.19.2.2.8.1.1.17	TCP/IP	ibmTCPIPmvsMIB	R/O
ibmMvsUdpMcastRecvLinkAddr	1.3.6.1.4.1.2.6.19.2.2.8.2.1.4	TCP/IP	ibmTCPIPmvsMIB	R/O
snmpSetSerialNo	1.3.6.1.6.3.1.1.6.1	Agent	RFC1907	R/O
snmpEngineID	1.3.6.1.6.3.10.2.1.1	Agent	RFC2271	R/O

Object Descriptor	Object Identifier	Supported by	Defined by	AA
snmpEngineBoots	1.3.6.1.6.3.10.2.1.2	Agent	RFC2271	R/O
snmpEngineTime	1.3.6.1.6.3.10.2.1.3	Agent	RFC2271	R/O
snmpEngineMaxMessageSize	1.3.6.1.6.3.10.2.1.4	Agent	RFC2271	R/O
snmpUnknownSecurityModels	1.3.6.1.6.3.11.2.1.1	Agent	RFC2272	R/O
snmpInvalidMsgs	1.3.6.1.6.3.11.2.1.2	Agent	RFC2272	R/O
snmpUnknownPDUHandlers	1.3.6.1.6.3.11.2.1.3	Agent	RFC2272	R/O
snmpTargetSpinLock	1.3.6.1.6.3.12.1.1	Agent	RFC2273	R/W
snmpTargetAddrTable	1.3.6.1.6.3.12.1.2	Agent	RFC2273	N/A
snmpTargetAddrEntry	1.3.6.1.6.3.12.1.2.1	Agent	RFC2273	N/A
snmpTargetAddrName	1.3.6.1.6.3.12.1.2.1.1	Agent	RFC2273	N/A
snmpTargetAddrTDomain	1.3.6.1.6.3.12.1.2.1.2	Agent	RFC2273	R/C
snmpTargetAddrTAddress	1.3.6.1.6.3.12.1.2.1.3	Agent	RFC2273	R/C
snmpTargetAddrTimeout	1.3.6.1.6.3.12.1.2.1.4	Agent	RFC2273	R/C
snmpTargetAddrRetryCount	1.3.6.1.6.3.12.1.2.1.5	Agent	RFC2273	R/C
snmpTargetAddrTagList	1.3.6.1.6.3.12.1.2.1.6	Agent	RFC2273	R/C
snmpTargetAddrParams	1.3.6.1.6.3.12.1.2.1.7	Agent	RFC2273	R/C
snmpTargetAddrStorageType	1.3.6.1.6.3.12.1.2.1.8	Agent	RFC2273	R/C
snmpTargetAddrRowStatus	1.3.6.1.6.3.12.1.2.1.9	Agent	RFC2273	R/C
snmpTargetParamsTable	1.3.6.1.6.3.12.1.3	Agent	RFC2273	N/A
snmpTargetParamsEntry	1.3.6.1.6.3.12.1.3.1	Agent	RFC2273	N/A
snmpTargetParamsName	1.3.6.1.6.3.12.1.3.1.1	Agent	RFC2273	N/A
snmpTargetParamsMPModel	1.3.6.1.6.3.12.1.3.1.2	Agent	RFC2273	R/C
snmpTargetParamsSecurityModel	1.3.6.1.6.3.12.1.3.1.3	Agent	RFC2273	R/C
snmpTargetParamsSecurityName	1.3.6.1.6.3.12.1.3.1.4	Agent	RFC2273	R/C
snmpTargetParamsSecurityLevel	1.3.6.1.6.3.12.1.3.1.5	Agent	RFC2273	R/C
snmpTargetParamsStorageType	1.3.6.1.6.3.12.1.3.1.6	Agent	RFC2273	R/C
snmpTargetParamsRowStatus	1.3.6.1.6.3.12.1.3.1.7	Agent	RFC2273	R/C
snmpUnavailableContexts	1.3.6.1.6.3.12.1.4	Agent	RFC2273	R/O
snmpUnknownContexts	1.3.6.1.6.3.12.1.5	Agent	RFC2273	R/O
snmpNotifyTable	1.3.6.1.6.3.13.1.1	Agent	RFC2273	N/A
snmpNotifyEntry	1.3.6.1.6.3.13.1.1.1	Agent	RFC2273	N/A
snmpNotifyName	1.3.6.1.6.3.13.1.1.1.1	Agent	RFC2273	N/A
snmpNotifyTag	1.3.6.1.6.3.13.1.1.1.2	Agent	RFC2273	R/C
snmpNotifyType	1.3.6.1.6.3.13.1.1.1.3	Agent	RFC2273	R/C

Object Descriptor	Object Identifier	Supported by	Defined by	AA
snmpNotifyStorageType	1.3.6.1.6.3.13.1.1.1.4	Agent	RFC2273	R/C
snmpNotifyRowStatus	1.3.6.1.6.3.13.1.1.1.5	Agent	RFC2273	R/C
usmStatsUnsupportedSecLevels	1.3.6.1.6.3.15.1.1.1	Agent	RFC2274	R/O
usmStatsNotInTimeWindows	1.3.6.1.6.3.15.1.1.2	Agent	RFC2274	R/O
usmStatsUnknownUserNames	1.3.6.1.6.3.15.1.1.3	Agent	RFC2274	R/O
usmStatsUnknownEngineIDs	1.3.6.1.6.3.15.1.1.4	Agent	RFC2274	R/O
usmStatsWrongDigests	1.3.6.1.6.3.15.1.1.5	Agent	RFC2274	R/O
usmStatsDecryptionErrors	1.3.6.1.6.3.15.1.1.6	Agent	RFC2274	R/O
usmUserSpinLock	1.3.6.1.6.3.15.1.2.1	Agent	RFC2274	R/W
usmUserTable	1.3.6.1.6.3.15.1.2.2	Agent	RFC2274	N/A
usmUserEntry	1.3.6.1.6.3.15.1.2.2.1	Agent	RFC2274	N/A
usmUserEngineID	1.3.6.1.6.3.15.1.2.2.1.1	Agent	RFC2274	N/A
usmUserName	1.3.6.1.6.3.15.1.2.2.1.2	Agent	RFC2274	N/A
usmUserSecurityName	1.3.6.1.6.3.15.1.2.2.1.3	Agent	RFC2274	R/O
usmUserCloneFrom	1.3.6.1.6.3.15.1.2.2.1.4	Agent	RFC2274	R/C
usmUserAuthProtocol	1.3.6.1.6.3.15.1.2.2.1.5	Agent	RFC2274	R/C
usmUserAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.6	Agent	RFC2274	R/C
usmUserOwnAuthKeyChange	1.3.6.1.6.3.15.1.2.2.1.7	Agent	RFC2274	R/C
usmUserPrivProtocol	1.3.6.1.6.3.15.1.2.2.1.8	Agent	RFC2274	R/C
usmUserPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.9	Agent	RFC2274	R/C
usmUserOwnPrivKeyChange	1.3.6.1.6.3.15.1.2.2.1.10	Agent	RFC2274	R/C
usmUserPublic	1.3.6.1.6.3.15.1.2.2.1.11	Agent	RFC2274	R/C
usmUserStorageType	1.3.6.1.6.3.15.1.2.2.1.12	Agent	RFC2274	R/C
usmUserStatus	1.3.6.1.6.3.15.1.2.2.1.13	Agent	RFC2274	R/C
vacmContextTable	1.3.6.1.6.3.16.1.1	Agent	RFC2275	N/A
vacmContextEntry	1.3.6.1.6.3.16.1.1.1	Agent	RFC2275	N/A
vacmContextName	1.3.6.1.6.3.16.1.1.1.1	Agent	RFC2275	R/O
vacmSecurityToGroupTable	1.3.6.1.6.3.16.1.2	Agent	RFC2275	N/A
vacmSecurityToGroupEntry	1.3.6.1.6.3.16.1.2.1	Agent	RFC2275	N/A
vacmSecurityModel	1.3.6.1.6.3.16.1.2.1.1	Agent	RFC2275	N/A
vacmSecurityName	1.3.6.1.6.3.16.1.2.1.2	Agent	RFC2275	N/A
vacmGroupName	1.3.6.1.6.3.16.1.2.1.3	Agent	RFC2275	R/C
vacmSecurityToGroupStorageType	1.3.6.1.6.3.16.1.2.1.4	Agent	RFC2275	R/C
vacmSecurityToGroupStatus	1.3.6.1.6.3.16.1.2.1.5	Agent	RFC2275	R/C

Object Descriptor	Object Identifier	Supported by	Defined by	AA
vacmAccessTable	1.3.6.1.6.3.16.1.4	Agent	RFC2275	N/A
vacmAccessEntry	1.3.6.1.6.3.16.1.4.1	Agent	RFC2275	N/A
vacmAccessContextPrefix	1.3.6.1.6.3.16.1.4.1.1	Agent	RFC2275	N/A
vacmAccessSecurityModel	1.3.6.1.6.3.16.1.4.1.2	Agent	RFC2275	N/A
vacmAccessSecurityLevel	1.3.6.1.6.3.16.1.4.1.3	Agent	RFC2275	N/A
vacmAccessContextMatch	1.3.6.1.6.3.16.1.4.1.4	Agent	RFC2275	R/C
vacmAccessReadViewName	1.3.6.1.6.3.16.1.4.1.5	Agent	RFC2275	R/C
vacmAccessWriteViewName	1.3.6.1.6.3.16.1.4.1.6	Agent	RFC2275	R/C
vacmAccessNotifyViewName	1.3.6.1.6.3.16.1.4.1.7	Agent	RFC2275	R/C
vacmAccessStorageType	1.3.6.1.6.3.16.1.4.1.8	Agent	RFC2275	R/C
vacmAccessStatus	1.3.6.1.6.3.16.1.4.1.9	Agent	RFC2275	R/C
vacmViewSpinLock	1.3.6.1.6.3.16.1.5.1	Agent	RFC2275	R/W
vacmViewTreeFamilyTable	1.3.6.1.6.3.16.1.5.2	Agent	RFC2275	N/A
vacmViewTreeFamilyEntry	1.3.6.1.6.3.16.1.5.2.1	Agent	RFC2275	N/A
vacmViewTreeFamilyViewName	1.3.6.1.6.3.16.1.5.2.1.1	Agent	RFC2275	N/A
vacmViewTreeFamilySubtree	1.3.6.1.6.3.16.1.5.2.1.2	Agent	RFC2275	N/A
vacmViewTreeFamilyMask	1.3.6.1.6.3.16.1.5.2.1.3	Agent	RFC2275	R/C
vacmViewTreeFamilyType	1.3.6.1.6.3.16.1.5.2.1.4	Agent	RFC2275	R/C
vacmViewTreeFamilyStorageType	1.3.6.1.6.3.16.1.5.2.1.5	Agent	RFC2275	R/C
vacmViewTreeFamilyStatus	1.3.6.1.6.3.16.1.5.2.1.6	Agent	RFC2275	R/C

Appendix D. NetView sample configuration

The following picture shows the configuration used for this redbook. Our machine was logically partitioned into three systems, RA03, RA28 and RA39. We used only one of these partitions and we did not implement any cross-domain function.

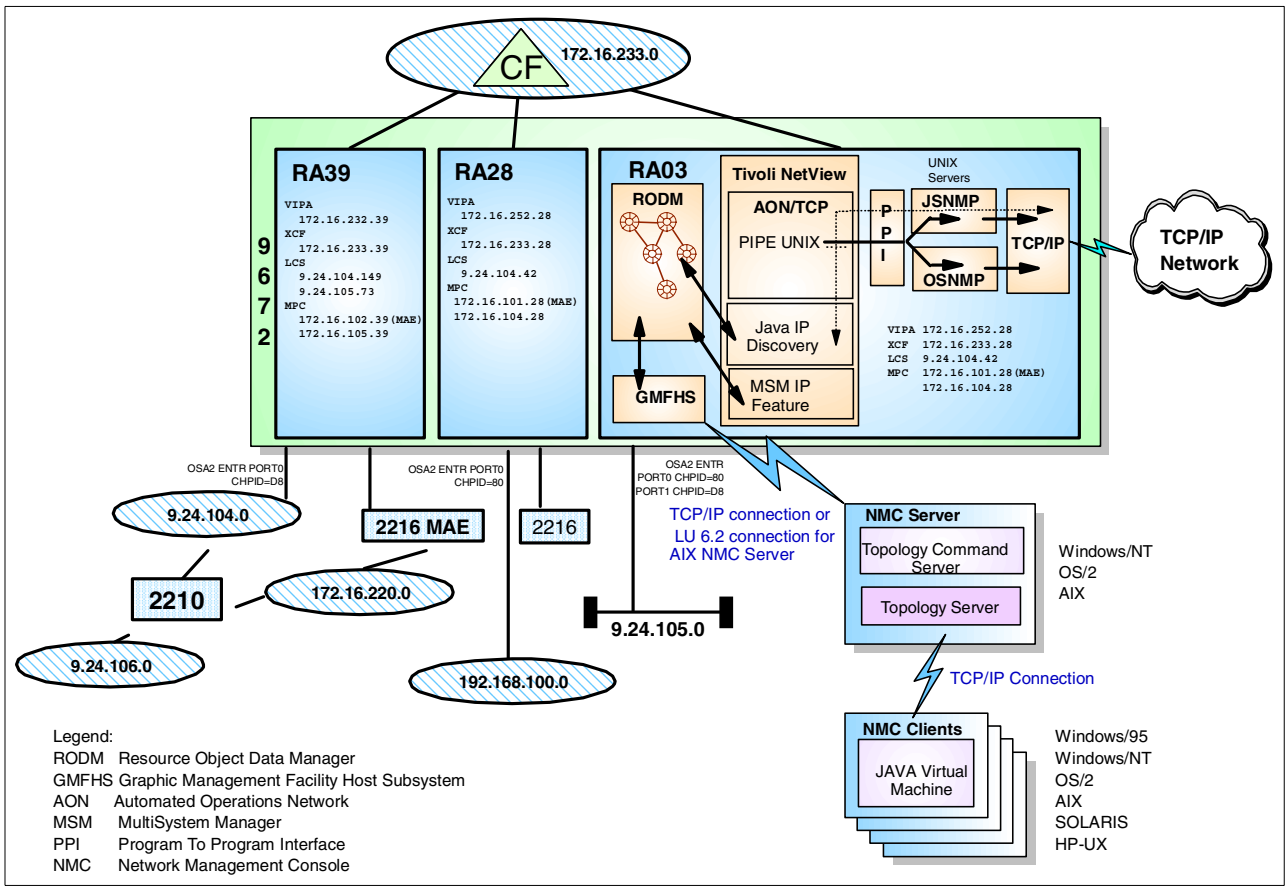


Figure 191. ITSO system configuration

On our system, the RA03 partition, we used the following products:

- OS/390 V2R8
- SecureWay Communication Server for OS/390 V2R8
- Tivoli NetView for OS/390 V1R3 with:
  - RODM
  - GMFHS

We also used a graphical workstation to install our NetView Management Console (NMC) with these characteristics:

1. Pentium processor

   The minimum processor required is one that can achieve a SPECInt95 score of at least 8.50. Most manufacturers provide the SPECInt95 scores for each of their machines. Additionally, a very large listing of SPECInt95 scores for a variety of manufacturers' machines is available on the Web at:

http://www.spec.org/osg/cpu95/results/cint95.html

For example, a Pentium Pro 200 MHz with 512 KB L2 cache or a Pentium II 233 MHz would qualify, 128 MB memory, 12 MB of additional fixed disk space for the NetView Management Console, screen resolution of 1024 by 768 with 256 colors

2. Windows/NT 4.0 (Windows NT 4.0 must be at service pack 3 level)
3. IBM Communications Server for Windows NT for LU netconv sessions.

   Personal Communications contains the emulators but does not have sufficient stress characteristics to support the NetView Management Console Server and problems reported using Personal Communications for server support will not be accepted by IBM. The NetView Management Console Server must be installed in an NTFS partition.

For further details on hardware and software prerequisites of NetView NMC please refer to the following URL: http://www.tivoli.com/nv390_supported.

D.1 OS/390 system environment

In this section we will describe general information on our OS/390 system, such as data set locations and RACF definitions.

D.1.1 NetView data sets

In this redbook we described every data set or member with the following naming convention:

DATASETNAME(MEMBER)

The following table shows the nicknames and the real names of each data set used in our configuration; please refer to it for every data set name we use in this book.

Table 27. Data set nicknames

Name	Real dataset name
STDENV	NETVIEW.V1R3USER.TCPPARMS
SYSTCPD	TCPIP.TCPPARMS
DSICLDU	NETVIEW.V1R3USER.RA03N.CNMCLST
DSIPARM	NETVIEW.V1R3M0.DSIPARM
DSIPARMU	NETVIEW.V1R3USER.RA03N.DSIPARM
DSIPRF	NETVIEW.V1R3M0.DSIPRF
DSIPRFU	NETVIEW.V1R3USER.RA03N.DSIPRF
INSTALL	NETVIEW.V1R3USER.INSTALL
EZLINST	NETVIEW.V1R3M0.SEZLINST
PROCLIB	USER.PROCLIB
EZLPNLU	NETVIEW.V1R3USER.RA03N.SEZLPNLU

D.1.2 NetView operator definitions

In our implementation we defined four new NetView operators as follows:

1. In member DSIPARM(DSIOPF) we uncommented the following statement:

```
%INCLUDE DSIOPFU
```

2. We added the following statements to member DSIDMN(DSIOPFU):

```
LUCY          OPERATOR    PASSWORD=LUCY
              PROFILEN    DSIPROFB
KAKKY         OPERATOR    PASSWORD=KAKKY
              PROFILEN    DSIPROFB
MARTIN        OPERATOR    PASSWORD=MARTIN
              PROFILEN    DSIPROFB
WOZA          OPERATOR    PASSWORD=WOZA
              PROFILEN    DSIPROFB
```

   The user profile DSIPROFB has been defined in DSIPRF(DSIPROFB) as shown below.

```
DSIPROFB      PROFILE  IC=LOGPROF1
              AUTH    MSGRECVR=YES,CTL=GLOBAL,NGMFADMN=YES
              OPCLASS 1,2
                END
```

D.1.3 RACF definitions

In order to use the UNIX services from NetView, all these RACF definitions are required:

1. Define the NetView procedure name to RACF with the following command:

```
RDEFINE STARTED NETVA03.* STDATA(user(tcpip3) group(omvsgrp) trusted(no))
setropts raclist(started) refresh
```

2. Define every NetView operator that will issue SNMP commands as RACF superusers. The RACF command to define an operator as Superuser is the following:

```
ADDUSER LUCY DFLTGRP(omvsgrp) OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
```

3. Define, also, the automated operators defined with the AUTOOPS statement in member DSIPARMU(FKXCFG01) as follows:

```
ADDUSER AUTTCP1 DFLTGRP(WTCRES) OMVS(UID(1001) HOME('/tmp')
PROGRAM('/bin/sh'))
ADDUSER AUTTCP2 DFLTGRP(WTCRES) OMVS(UID(1001) HOME('/tmp')
PROGRAM('/bin/sh'))
ADDUSER AUTTCP3 DFLTGRP(WTCRES) OMVS(UID(1001) HOME('/tmp')
PROGRAM('/bin/sh'))
.....
```

4. Define auto-operator AUTODIS1 as follows:

```
ADDUSER AUTODIS1 DFLTGRP(OMVSGRP) OMVS(UID(0) HOME('/') PROGRAM('/bin/sh'))
```

5. If the UNIX command server (see Appendix D.4, “UNIX command server installation” on page 253) will be started as a started tasks, define the following:

```
rdefine started CNMSSUNX.* stdata(user(tcpip3) group(omvsgrp) trusted(no))
setropts raclist(started) refresh
```

To display the NetView operator definition in the RACF user profile, issue the following command:

```
lu lucy omvs noracf
```

The output should be like this:

```
USER=LUCY
OMVS INFORMATION
----------------
UID= 0000000000
HOME= /u/lucy
PROGRAM= /bin/sh
CPUTIMEMAX= NONE
ASSIZEMAX= NONE
FILEPROCMAX= NONE
PROCUSERMAX= NONE
THREADSMAX= NONE
MMAPAREAMAX= NONE
```

D.1.4 RODM

The procedure used to start the RODM address space is provided as a sample and is copied in the PROCLIB data set during product installation. The member name is EKGXRODM. During startup it reads member NETVIEW.V1R3M0.SEKGSMP1(EKGCUST) where its environment parameters are defined. The only parameter we have modified in this installation was the SEC_CLASS name. We used the value (*TSTRODM) to bypass the security checking for the RODM access levels. See Appendix E.7, “RODM configuration files” on page 294 for member sample.

D.1.5 GMFHS

The procedure name for the GMFHS address space is called CNMGMFHS and is copied in the PROCLIB data set during installation. The GMFHS procedure reads member DSIPARMU(DUIGINIT) during startup and tries to connect to RODM using the name found in the RODMNAME parameter. You need to change this parameter in order to reflect the exact RODM name (the name provided with the NAME parameter in the RODM start command). Change also the DOMAIN parameter to reflect your NetView domain name as in the DOMAINID keyword in DSIPARMU(DSIDMNK). See Appendix E.8, “GMFHS configuration files” on page 301 for member sample.

D.2 NetView AON/TCP installation

This section provides the NetView installation and configuration information, including the AON/TCP information.

D.2.1 NetView started task

In the NetView Startup Procedure CNMPROC (sample name is CNMSJ009) ensure that the following data sets are uncommented:

```
//CNMAPROC PROC
//NETVIEW  EXEC PGM=BNJLINTX,TIME=1440,
//         REGION=50M,
//         PARM=(24K,200,'RA03N','','*NOARM',''),
```

```
//         DPRTY=(13,13)
//STEPLIB  DD   DSN=NETVIEW.V1R3M0.CNMLINK,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEKGMOD1,DISP=SHR
//         DD   DSN=REXX.V1R3M0.SEAGLMD,DISP=SHR
//         DD   DSN=REXX.V1R3M0.SEAGALT,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEZLLINK,DISP=SHR 1
//SYSTCPD  DD DSN=TCPIP.TCPPARMS(TCPD&SYSCLONE.A),DISP=SHR
//DSICLD   DD   DSN=NETVIEW.V1R3USER.RA03N.CNMCLST,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.CNMCLST,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEZLCLST,DISP=SHR 2
//         DD   DSN=NETVIEW.V1R3M0.CNMSAMP,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEKGSMP1,DISP=SHR
//DSIOPEN  DD   DSN=NETVIEW.V1R3USER.RA03N.SDSIOPEN,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SDSIOPEN,DISP=SHR
//DSIPARM  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIPARM,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.DSIPARM,DISP=SHR
//DSILIST  DD   DSN=NETVIEW.V1R3USER.RA03N.DSILIST,DISP=SHR
//DSIASRC  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIASRC,DISP=SHR
//DSIARPT  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIARPT,DISP=SHR
//DSIVTAM  DD   DSN=NETVIEW.V1R3USER.RA03N.VTAMLST,DISP=SHR
//         DD   DSN=SYS1.VTAMLST,DISP=SHR
//DSIPRF   DD   DSN=NETVIEW.V1R3USER.RA03N.DSIPRF,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.DSIPRF,DISP=SHR
//DSIMSG   DD   DSN=NETVIEW.V1R3M0.SDSIMSG1,DISP=SHR
//BNJPNL1  DD   DSN=NETVIEW.V1R3M0.BNJPNL1,DISP=SHR
//BNJPNL2  DD   DSN=NETVIEW.V1R3M0.BNJPNL2,DISP=SHR
//CNMPNL1  DD   DSN=NETVIEW.V1R3M0.CNMPNL1,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEKGPNL1,DISP=SHR
//         DD   DSN=NETVIEW.V1R3USER.RA03N.SEZLPNLU,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEZLPNLU,DISP=SHR 3
//DSILOGP  DD   DSN=NETVIEW.V1R3USER.RA03N.DSILOGP,
//         DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSILOGS  DD   DSN=NETVIEW.V1R3USER.RA03N.DSILOGS,
//         DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSITRCP  DD   DSN=NETVIEW.V1R3USER.RA03N.DSITRCP,
//         DISP=SHR,AMP=AMORG
//DSITRCS  DD   DSN=NETVIEW.V1R3USER.RA03N.DSITRCS,
//         DISP=SHR,AMP=AMORG
//AAUVSPL  DD   DSN=NETVIEW.V1R3USER.RA03N.AAUVSPL,
//         DISP=SHR,AMP='AMORG'
//AAUVSSL  DD   DSN=NETVIEW.V1R3USER.RA03N.AAUVSSL,
//         DISP=SHR,AMP='AMORG'
//BNJLGPR  DD   DSN=NETVIEW.V1R3USER.RA03N.BNJLGPR,
//         DISP=SHR,AMP='AMORG'
//BNJLGSE  DD   DSN=NETVIEW.V1R3USER.RA03N.BNJLGSE,
//         DISP=SHR,AMP='AMORG'
//DSIKPNL  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIKPNL,
//         DISP=SHR,AMP=AMORG
//DSISVRT  DD   DSN=NETVIEW.V1R3USER.RA03N.DSISVRT,
//         DISP=SHR,AMP=AMORG
//EZLSTAT  DD   DSN=NETVIEW.V13USER.RA03N.STATS, 4
//         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//EZLLOGP  DD   DSN=NETVIEW.V13USER.RA03N.LOGP, 5
//         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//EZLLOGS  DD   DSN=NETVIEW.V13USER.RA03N.LOGS, 6
//         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//SYSPRINT DD   SYSOUT=*
```

1 Modules data set
2 CLIST data set
3 Panel data set
4 Status VSAM data set
5 , 6 LOG VSAM data set

D.2.2 Copying AON members to DSIPARM and DSIPRF

AON requires definition files that must be copied to NetView data sets. To do this, run the EZLINST(EZLSJ100) job. This job copies the required AON definitions into DSIPARMU, DSIPRFU, and EZLPNLU.

Notes: Copy the sample in a user data set and check if the names in it will match your environment.

D.2.3 Allocating the automation log file and status file data sets

To allocate the automation log file and status file data sets, run the EZLINST(EZLSJ008) job. This job defines the VSAM clusters used by various AON components. The following table lists the names of the components, the names of their data sets, and the names of the members that contain the VSAM cluster information for those data sets.

Table 28. Installation jobs for the VSAM clusters used by Tivoli NetView for OS/390

Step	Member	Purpose	Data sets being allocated
STEP1	EZLSID01	Deletes all VSAM databases	none
STEP2	EZLSI101	Allocates the status file data set	NETVIEW.V13USER.RA03N.STATS
STEP3	EZLSI201	Allocates the log file data sets	NETVIEW.V13USER.RA03N.LOGP NETVIEW.V13USER.RA03N.LOGS

Note: Check the member's contents to define the correct volume and data set names.

D.2.4 Adding command model statements

Edit DSIPARMU(DSICMDM) and uncomment the following statements:

```
%INCLUDE EZLCMD (base AON statements)
%INCLUDE FKXCMD (AON/TCP)
```

The following is an example of how the %INCLUDE statement works:

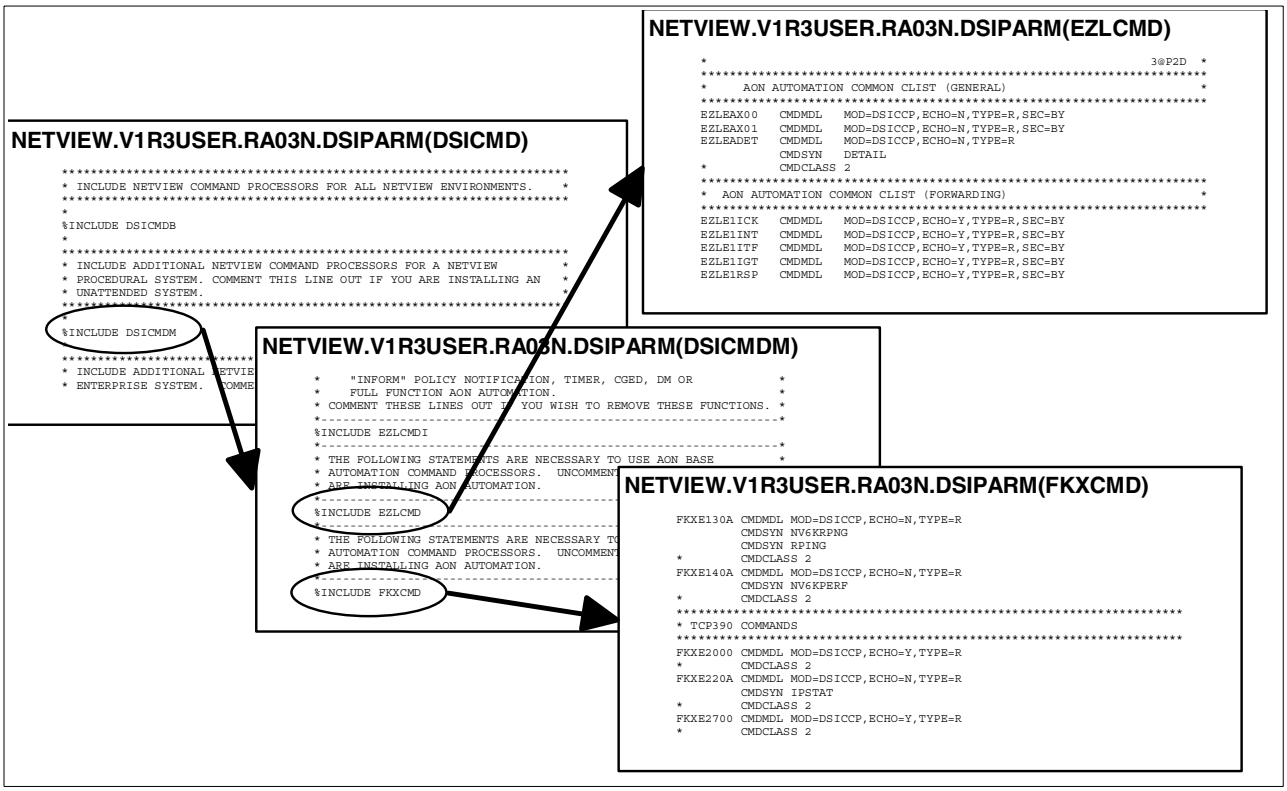


Figure 192. %INCLUDE statement

D.2.5 Updating the AON control file

Edit the DSIPARMU(EZLCFG01) member and uncomment the following statement:

```
%INCLUDE FKXCFG01 (AON/TCP)
```

D.2.6 Customizing the automation table

The default automation table shipped with NetView is DSIPARM(DSITBL01). The statements to include AON automation are commented out and must be uncommented for automation to occur.

Copy the DSIPARM(DSITBL01) in the DSIPARMU data set and remove the comments from the following statements to enable the AON base automation toolkit:

```
%INCLUDE EZLMSUST
%INCLUDE EZLDWOAO

IF TEXT=.'DSISVRT'. THEN
EXEC(CMD('CNME8003 COMMON '));
IF TEXT=.'EZLTLOG'. THEN
EXEC(CMD('EZLEALRS ' MESSAGE));

IF MSGID='DSI546I'| MSGID='DSI547I' THEN
BEGIN;
IF TOKEN(2)='EZLTLOG' THEN
EXEC(CMD('EZLEALSW' MESSAGE) ROUTE(ALL *)) HOLD(Y);
```

```
IF TOKEN(2)&ne.'EZLTLOG' THEN HOLD(Y);
END;
IF MSUSEG(0000.10.11.07 3)='TECUPD' THEN
SRF(TECROUTE PASS);
%INCLUDE EZLDSI24
%INCLUDE EZLDSIAO
%INCLUDE EZLEKGAO
%INCLUDE EZLTBL01
%INCLUDE EZLTBL02
%INCLUDE EZLTBL03
```

Remove the comments from the following statements in DSITBL01 to enable AON/TCP:

```
%INCLUDE FKXTBL01
%INCLUDE FKXMSU01
```

Note: When you uncomment FKXMSU01, browse DSITBL01 and its other included members (for example, FLCSTBLI) for an overlap. If there is an overlap, code a CONTINUE(Y) statement in the member that appears first. Use this only if you are running TCP/IP for MVS, and you want to convert SNMP traps to alerts and have them processed by AON.

Test the automation table by entering the following command at a running NetView where the automation table is in the DSIPARM data set:

```
AUTOTBL MEMBER=member,TEST
```

Notes:

1. Renumber the NetView automation table after editing it. When the NetView automation table processes a message and finds a match that results in a command or command list being run, it writes NetView message CNM493I to the NetView log file. Message CNM493I contains the line number of the automation table entry matched.
2. If you want to use an automation table name other than DSITBL01 or a control file name other than EZLCFG01, you must edit EZLDSI24 accordingly:

```
* INITIALIZE AON IF NOT ALREADY ACTIVE
             IF MSGID='DSI240I' & TEXT=.'NETWORK LOG IS NOW ACTIVE'.
                & DOMAINID=%NV_DOMAIN%
                & ATF('DSICGLOB EZLINIT.AON') &ne. 'Y'.
               THEN
              EXEC(CMD('EZLEANTL CFG=EZLCFG01,MSG=DSITBL01')
              ROUTE(ONE AUTO1));
```

   In this statement, change the EZLCFG01 in CFG=EZLCFG01 to the name of the AON control file you want loaded. Change the DSITBL01 in MSG=DSITBL01 to the name of the automation table that you want loaded.
3. You can turn off CNM493I messages with the NetView DEFAULTS command.
4. If you want to browse the automation table with the %INCLUDEs resolved, use the following command from the command facility display (NCCF) in NetView:

   ```
   BR DSITBL01
   ```
5. The automation table load process also generates a listing file called DSITBTST. You can browse the listing to see %INCLUDEs and synonyms resolved.

6. The AON automation table is loaded during AON initialization and by the AON operator on demand (AON 1.8.2 or AONINIT from the command line). AON replaces the currently active member with the member specified as follows:
   a. If an AON automation table is not loaded, AON attempts to load its table 1 (AT=1). If a non-AON table is already loaded as FIRST, the AON table is loaded in the second position (AT=2).
   b. If an AON table is currently active, the new table is swapped with the currently loaded table (table position is retained).
   c. If a non-AON table is currently active with the same name as the new AON table being loaded, the existing AON table is unloaded and the new table is swapped with the table of the same name.

D.2.7 Adding operator definitions and passwords

Edit DSIPARMU(DSIOPFM) and uncomment the following statements:

```
%INCLUDE EZLOPF
%INCLUDE FKXOPF (AON/TCP)
```

D.2.8 Changing task statements

Edit DSIPARMU(DSIDMNM) member and uncomment the following statement:

```
%INCLUDE EZLDMN
```

Note: Make sure that you define enough VTAM APPL statements the NetView ACB definitions to allow for a sufficient number of operators and autotasks to be logged on.

D.2.9 Online help

Move member DSIPARM(CNMHELPM) to the DSIPARMU data set and uncomment the following line:

```
%INCLUDE FKXHELP
```

This member contains the NetView mapping table for online help.

D.2.10 FKXTABLE member

Ensure that `EZLOPT IP390,ENABLE=Y` is coded in the DSIPARMU(FKXTABLE) member.

D.3 Installing MultiSystem manager

We will describe only the operations needed to install the NetView Resource Specific Command Sets on a Windows NT workstation. For other platforms, please refer to the FLCREAD1.ME (English) or FLCREAD3.ME (Japanese) files located in the /usr/lpp/netview/install directory (OS/390 UNIX System Services).

D.3.1 Installing the resource specific command sets

We downloaded the code from the following URL:

```
http://www.tivoli.com/nv390_supported
```

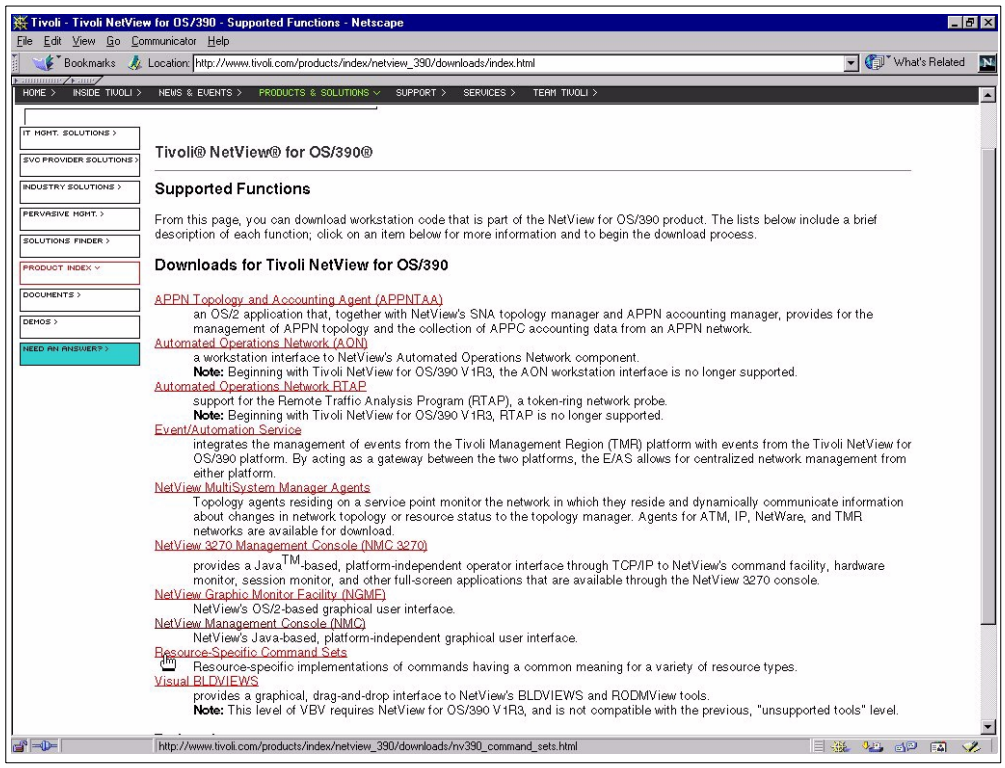


Figure 193. Tivoli NetView for OS/390 Supported Functions

Click **Resource-Specific Command Sets**:

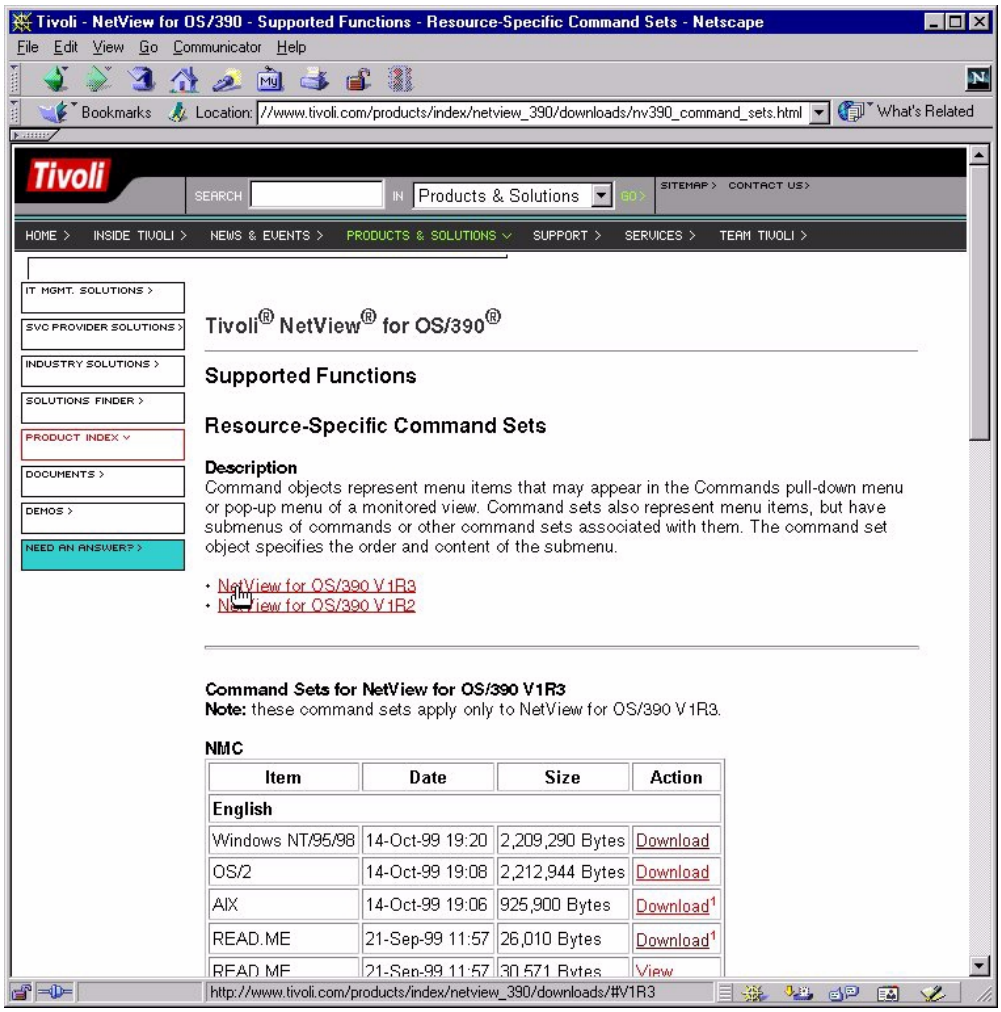


Figure 194. Tivoli NetView for OS/390 Supported Functions - Resource-Specific Command Sets

Click **NetView for OS/390 V1R3**:

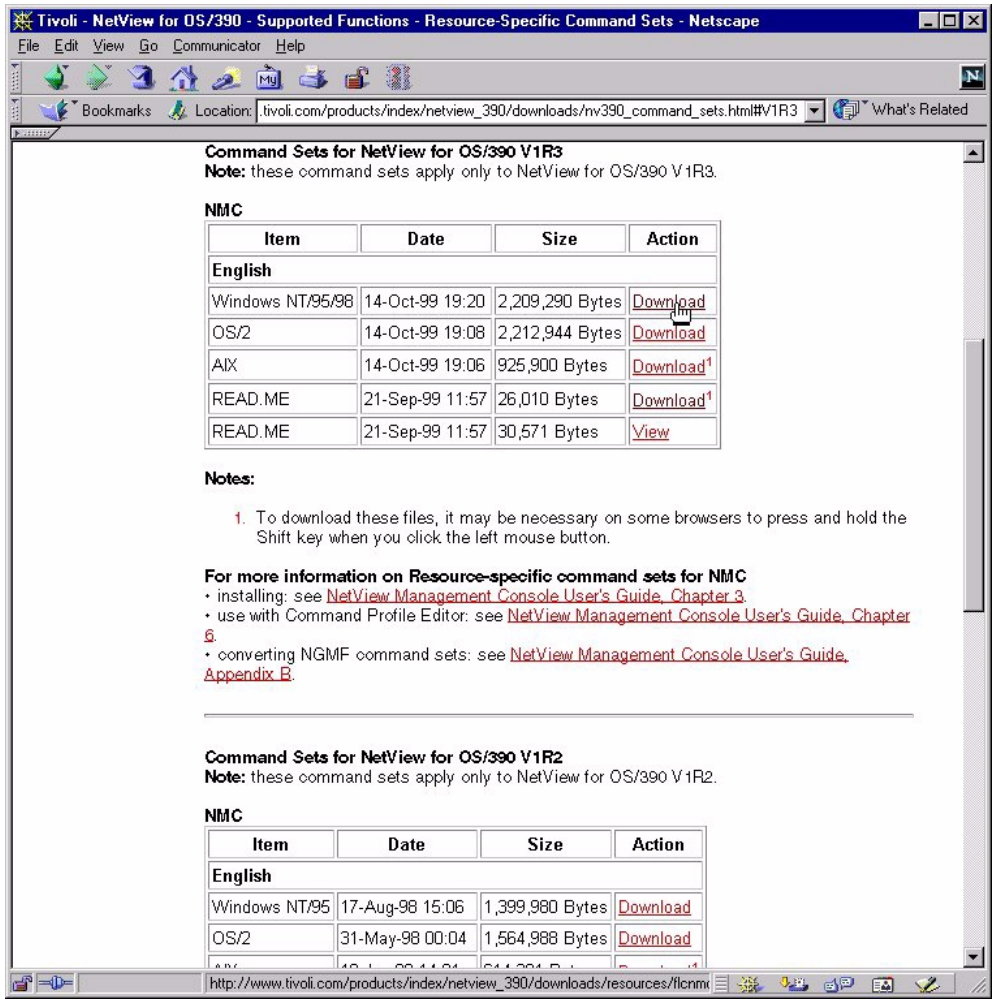


Figure 195. Tivoli NetView for OS/390 Supported Functions - downloading the code

Click **Download** at the right of Windows NT/95/98:

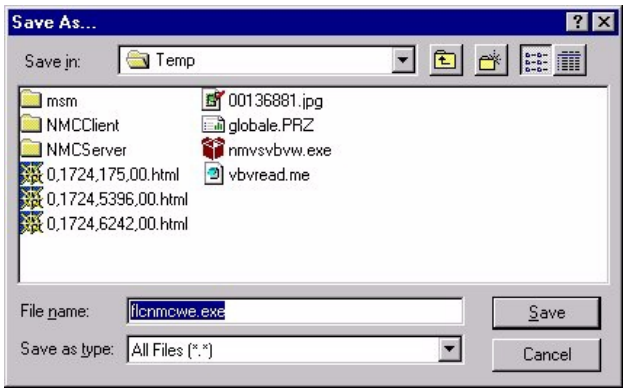


Figure 196. Saving the code

Download the Windows version of the NMC Resource Specific Command Sets into a temporary directory as flcnmcwe.exe.

D.4 UNIX command server installation

To facilitate the running of OS/390 UNIX commands initiated from NetView, a dedicated PPI receiver, CNMEUNIX, receives commands and data from NetView. A server process running in an OS/390 UNIX address space waits on this PPI receiver for incoming commands and data.

The server process is called CNMEUNIX and runs as an OS/390 UNIX kernel process. The OS/390 UNIX server consists of three parts which must be installed in the UNIX hierarchical file system (HFS). The default directory into which the installation will install the parts is /usr/lpp/netview/bin.

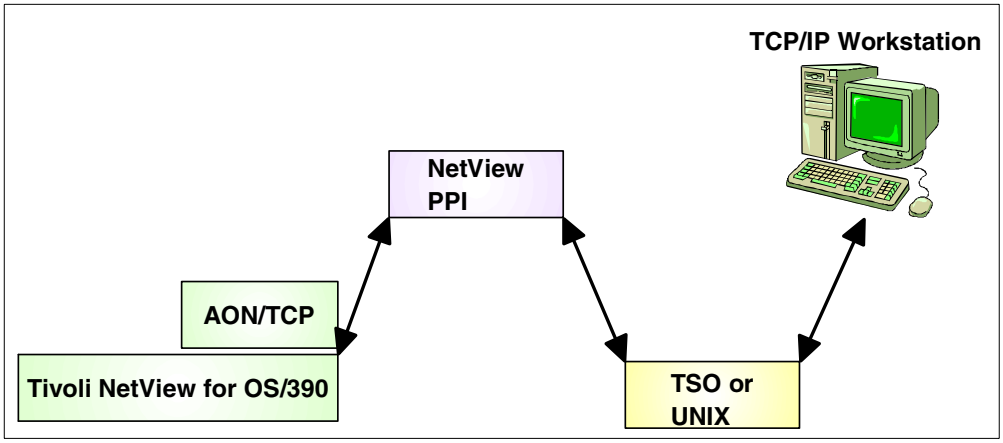


Figure 197. UNIX server overview

D.4.1 How to start the UNIX server

Basically, we have two ways to start the UNIX server: from NetView or from the UNIX shell. In our configuration we started the UNIX server from NetView as a submitted job. For a complete description of these procedures please refer to *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236.

Here is the description of our implementation:

The NetView command used to start the UNIX server is the following:

```
START UNIXSERV=*
```

Ensure that the sample job CNMSJUNX is contained in a DSIPARM data set. We used this member:

```
//CNMEUNIX JOB ,'NetView Unix Server',MSGCLASS=R
//        EXEC PGM=BPXBATCH,
//             PARM='PGM /usr/lpp/netview/bin/cnmeunix'
//STEPLIB   DD DISP=SHR,DSN=REXX.V1R3M0.SEAGALT
//STDOUT    DD PATH='/tmp/cnmeunix.stdout',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=SIRWXU
//STDERR    DD PATH='/tmp/cnmeunix.stderr',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=SIRWXU
//STDENV    DD DSN=NETVIEW.V1R3USER.TCPPARMS(AON03ENV),DISP=SHR
```

```
//STDOUT  EXEC PGM=IKJEFT01,COND=((256,LE),EVEN)
//SYSTSPRT  DD SYSOUT=*
//FROMHFS   DD PATH='/tmp/cnmeunix.stdout',
//             PATHOPTS=(ORDONLY,OCREAT)
//TOSYSOUT  DD SYSOUT=*,
//             RECFM=F,BLKSIZE=255
//SYSTSIN   DD DATA
         OCOPY INDD(FROMHFS)    +
               OUTDD(TOSYSOUT)  +
               TEXT             +
               CONVERT(YES)     +
               PATHOPTS(USE)
//STDERR  EXEC PGM=IKJEFT01,COND=((256,LE),EVEN)
//SYSTSPRT  DD SYSOUT=*
//FROMHFS   DD PATH='/tmp/cnmeunix.stderr',
//             PATHOPTS=(ORDONLY,OCREAT)
//TOSYSOUT  DD SYSOUT=*,
//             RECFM=F,BLKSIZE=255
//SYSTSIN   DD DATA
         OCOPY INDD(FROMHFS)    +
               OUTDD(TOSYSOUT)  +
               TEXT             +
CONVERT(YES)     +
               PATHOPTS(USE)
//
```

The STDENV(AON03ENV) contains lines greater than 80 characters so it has been allocated as (FB, lrecl=512, blksize=27648). It contains the following statements (the CLASSPATH line has been truncated because it is too long for the picture but in the real member it is on one line only):

```
RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD03A)'
PATH=/bin:/usr/lpp/netview/bin:/usr/lpp/tcpip/bin:/usr/lpp/java/J1.1/bin
MANPATH=/usr/lpp/netview/man/%L:/usr/man/%L
CLASSPATH=/usr/lpp/netview/lib/ihssnmps.jar:/usr/lpp/java/J1.1/lib/classes.zip:
/usr/lpp/netview/lib/ipdiscovery.jar
LIBPATH=/usr/lpp/netview/lib
LD_LIBRARY_PATH=/usr/lpp/netview/lib
TZ=EST5EDT
```

Figure 198. NETVIEW.V1R3USER.TCPPARMS(AON03ENV) member

If you are running AON/TCP, AON will automatically start the UNIX server based on the TCP390 stack definitions.

Ensure that the value of DEFAULT STRTSERV in NetView is set to SBMTJOB (default). To display the value of STRTSERV issue the following command from NCCF:

```
LIST DEFAULTS
```

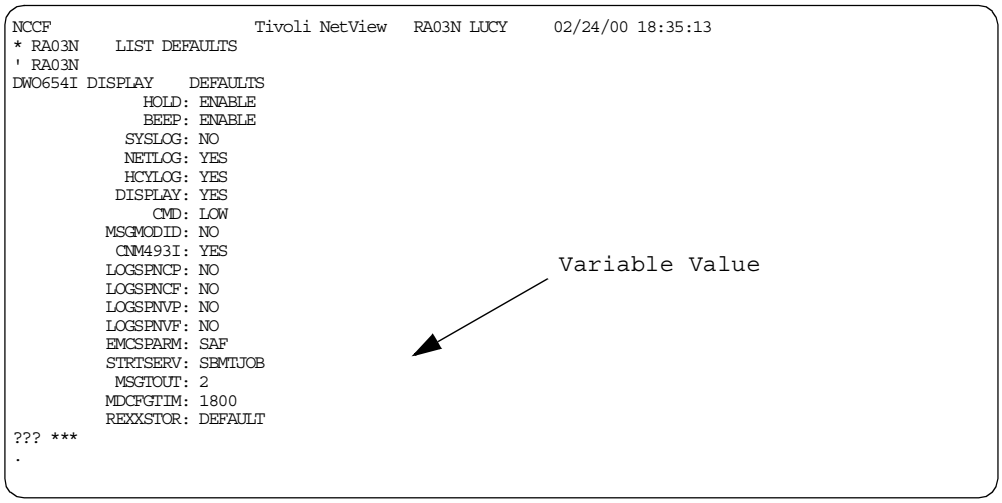


Figure 199. LIST DEFAULTS

If you need to change the DEFAULT STRTSERV value you can issue the command from a NetView NCCF panel. The syntax is the following:

```
                                +-=SBMTJOB--+
>>--DEFAULTS-----+--- STRTSERV-+-----------+-
                                +-=STRTPROC-+
```

Note: You can insert the command in the NetView initial CLIST (CNME1034) copying the CLIST in a user CLIST data set.

If multiple versions of the OS/390 UNIX command server JCL are required, the optional MEM parameter may be specified on the START UNIXSERV command to specify members other than the default which is CNMSJUNX for submitted jobs or CNMSSUNX for started tasks.

D.5 Install the Java SNMP services

In this chapter we will describe the functionality and the installation steps for the Java SNMP services (JSNMP). This function is required only if you plan to use the NMC MIB browser and the Java SNMP JTRAP command.

The following picture shows the JSNMP major functions:

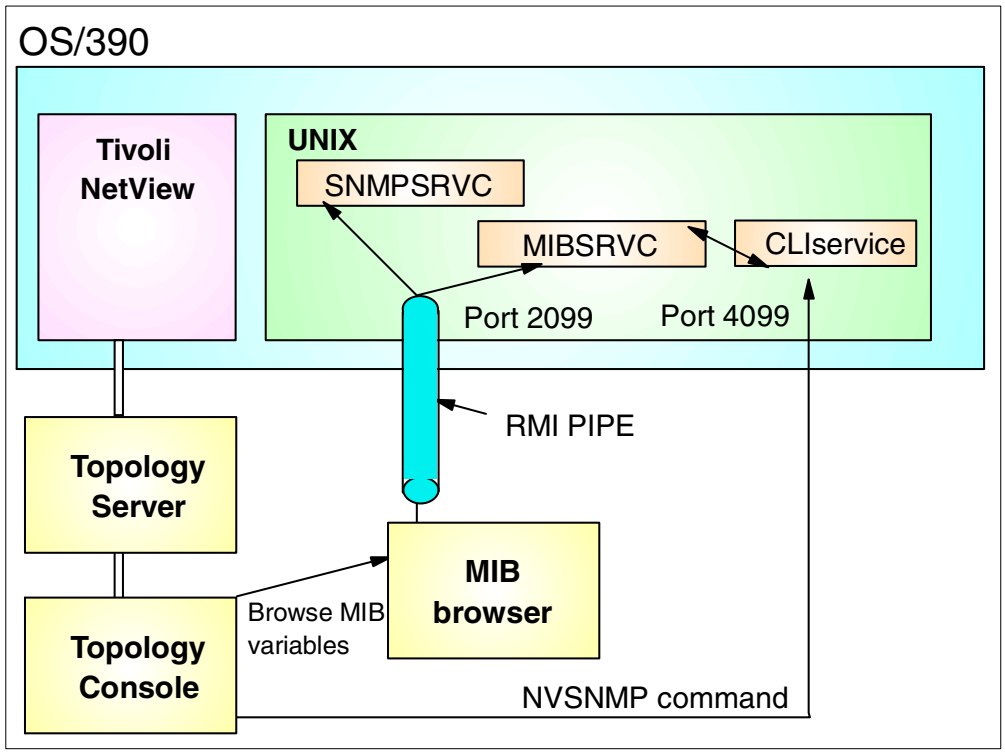


Figure 200. JSNMP function

Java SNMP services consists of MIBSRVC (MIB service) and SNMPSRVC (SNMP service). MIBSRVC provides MIB metadata information to the client. SNMPSRVC provides SNMP stack functions and it makes SNMP queries on behalf of clients. In this case the client can be a MIB browser from NMC or an NVSNMP Java command.

When SNMPSRVC is started, it brings up another service for the command line interface (CLI service).

MIBSRVC parses MIBs (management information base) and provides MIB metadata information to NVSNMP commands with Java or a MIB browser. It can parse MIBs during initialization by reading the file /etc/netview/nv390mibs.def.

The MIB browser is a client of MIBSRVC and connects to the host assigned to mibhost and the port assigned to mibport listed in the properties file $(bindir)/TDS/server/db/current/settings/defaultmibbr.properties on the NMC server.

SNMPSRVC is used by a MIB browser for SNMP commands. The NVSNMP command with Java is a client of SNMPSRVC and connects to the host assigned to srvhostname and the port assigned to srvport. The host and port assignment are listed in the configuration file /etc/netview/nv390srvr.conf.

SNMPSRVC then connects to MIBSRVC to retrieve MIB metadata information. SNMPSRVC connects to the host assigned to mibhost and the port assigned to mibport in the configuration file. MIBSRVC must be active with the appropriate -port option.

D.5.1 Installation

Follow the following steps to install the JSNMP stack:

D.5.1.1 Defining OS/390 UNIX shell variables

In the STDENV(AON03ENV) member modify the following statements:

- Add Java, NetView and TCP/IP bin to the PATH statement as follows:

  ```
  PATH=/bin:/usr/lpp/netview/bin:/usr/lpp/tcpip/bin:/usr/lpp/java/J1.1/bin
  ```

- Add CLASSPATH statement with ihssnmps.jar and Java classes.zip as shown in the following example:

  ```
  CLASSPATH=/usr/lpp/netview/lib/ihssnmps.jar:/usr/lpp/java/J1.1/lib/class
  es.zip:/usr/lpp/netview/lib/ipdiscovery.jar
  ```

- Add RESOLVER_CONFIG statement with TCPDATA as shown in the following example:

  ```
  RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD03A)'
  ```

- Add MANPATH as shown in the following example:

  ```
  MANPATH=/usr/lpp/netview/man/%L:/usr/man/%L
  ```

- Ensure that the JAVA_COMPILER variable is set to off if you are running JDK 1.1.8 on OS/390.

  ```
  JAVA_COMPILER=off
  ```

  This setting is required by the MIB browser function and is also documented in the PSP for NetView FMID JPZ8407. Under the following environments, the JAVA_COMPILER setting is not required:

  - If you are using JDK 1.1.6
  - If you are running JDK 1.1.8 dated 01/08/2000 or later

  In this case, you should remove the JAVA_COMPILER line from the environment variable configuration file, so that the JIT, which should provide some performance improvement, available in your implementation will run, because the default is to have the JIT active.

  The level of JDK can be displayed by the `java -fullversion` command from the OS/390 UNIX shell.

D.5.1.2 Defining configuration files

SNMPSRVC requires the configuration files snmp.conf and nv390srvr.conf. It refers to the snmp.conf file to communicate with the SNMP agent. The snmp.conf file is used when an SNMP query is made without a community string. For example, if you request a GET, it uses getCommunityName, and if you request a SET it uses the setCommunityName entry for the SNMP agent. For GET, GETNEXT, WALK, GETBULK and BULKWALK it will use getCommunityName.

Update your copy of nv390srvr.conf and snmp.conf to reflect the proper IP address for keywords mibhost, srvhostname and snmphost.

If you want to bring up SNMPSRVC on a different port, change srvport and snmpport. If you want to bring up MIBSRVC on a different port, change mibport.

The following are the configuration files used in our configuration; the files are in the /etc/netview/ directory:

```
srvhostname = 9.24.104.113
srvport = 4099
custdelimiter = *
mibhost = 9.24.104.113
mibport = 2099
snmphost = 9.24.104.113
snmpport = 2099
```

Figure 201. /etc/netview/nv390srvr.conf sample configuration

```
9.24.104.113:3s:3:30m:publicv1:5s:1:1d:publicv1:161:*
9.24.104.42:3s:3:30m:publicv1:5s:1:1d:publicv1:161:*
9.24.104.149:3s:3:30m:publiv1:5s:1:1d:publicv1:161:*
9.24.105.126:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
9.24.105.73:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.250.3:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.252.28:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.232.39:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
9.24.105.*:3s:3:30m:public:2s:1:1d:public:161:*
*.*.*.*:3s:3:30m:public:2s:1:1d:public:161:*
```

Figure 202. /etc/netview/snmp.conf sample configuration

MIBSRVC requires the MIB definition file nv390mibs.def during initialization. A minimum required set of SNMP MIBs are provided which are required by the MIB browser.

You can load the non-default MIB definitions by configuring the file names, in which the MIB objects are defined using the SMI syntax, in /etc/netview/nv390mibs.def.

The following is the MIB definition file used at ITSO Raleigh:

```
#*********************************************************************
# Licensed Materials - Property of Tivoli Systems                    *
# 5697-B82 (c) Copyright Tivoli Systems 1999                         *
# All rights reserved.                                               *
#                                                                    *
# US Government Users Restricted Rights - Use, duplication or        *
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  *
#*********************************************************************
#*********************************************************************
#**In this file # in first column makes it a comment line.        ***
#*********************************************************************
/usr/lpp/netview/mibs/rfc1213-MIB-II.mib
/usr/lpp/netview/mibs/rfc1903.mib
/etc/netview/tcpip/dpiSimpl.mib
/etc/netview/tcpip/rfc1592b1.mib
/etc/netview/tcpip/ibm3172.mib
/etc/netview/tcpip/saMIB.mib
/etc/netview/tcpip/ibm3172b.mib
/etc/netview/tcpip/saMIB2.mib
/etc/netview/tcpip/mvstcpip.mib
/etc/netview/tcpip/mvstcpip2.mib
/etc/netview/tcpip/rfc1592b.mib
```

Figure 203. /etc/netview/nv390mibs.def sample configuration

D.5.1.3 Copying the nv39srvr.conf file on the NMC server

Copy the updated file /etc/netview/nv390srvr.conf from UNIX System Services to your NMC servers. Copy the file to the following directory and use the following file name:

```
$(bindir)/TDS/server/db/current/settings/defaultmibbr.properties
```

D.5.1.4 Starting Java SNMP services

The command used to start the JSNMP services is:

```
netvasis nvsrvc start
```

The command produces the following output:

```
x nvsrvc is not running.
x nvsrvc starting....
x SMI MIB Metadata - registering ...
x SMI MIB Metadata - received java.rmi.ConnectException
x SMI MIB Metadata - registry not found or not accesible
x SMI MIB Metadata - starting new registry on port 2099
x SMI MIB Metadata - registered
x Number of arguments = 1
x Remotely processing file: /etc/netview/nv390mibs.def
x     Processing File: /etc/netview/nv390mibs.def
x     Processing File: /usr/lpp/netview/mibs/rfc1213-MIB-II.mib
x     Processing File: /usr/lpp/netview/mibs/rfc1903.mib
x Processing complete for file: /etc/netview/nv390mibs.def
x SMI MIB Metadata - initialization complete
x Attached to Mib Service.9.24.104.1132099
x Trying to create registry for SnmpCliService
x Registry created
x SnmpCliService object exported
x SNMP tracer service is ready.
x Snmp:InitialConfig: Definitions not loaded:java.io.FileNotFoundException: us
x   Defaults will be used.
x Snmp:InitialConfig: Definitions not loaded:java.io.FileNotFoundException: sy
x   Defaults will be used.
x Ready for rebind
x SnmpCliService bound in registry & ready to start
x SnmpService bound to registry
```

Figure 204. JSNMP services startup messages

D.6 Installing Java-based IP discovery agent

This section provides additional information on the Java-based IP discovery installation and configuration. Also refer to 5.7, “Java-based IP discovery” on page 156 for more information.

D.6.1 Parameters for Java-based IP discovery configuration file

There are two configuration files which need to be set up for IP discovery to work. The first, by default, is /etc/netview/ipdiscovery.conf. This file is used primarily to specify seed IP addresses for discovery to start from and the scope of addresses to limit discovery. The second file is, by default, /etc/netview/snmp.conf. This file specifies parameters to control how IP discovery performs pings and snmp polls.

IP discovery configuration file is used to specify IP discovery parameters. Each parameter is specified on a line by itself. Blank lines are ignored as are lines with a # in column 1 (comments). The format of a parameter is keyword=value.

Table 29 summarizes the parameters that can be configured in this file.

Table 29. Parameter specifications for the IP discovery configuration file

Keyword	Type	Default	Description
discoveryPollingInterval	Time interval	2h	Interval to rediscover new IP addresses. To turn IP discovery off, set this value to -1.
pingSpreadMask	IP Address	255.255.255.255	The mask used to decide which networks will be discovered by a ping spread instead of snmp polling. Networks with IP network address bits greater than this mask will be polled by ping spread. Otherwise, they will be polled by snmp.
pingSpreadInterval	Time interval	50ms	Time interval between pings of different IP addresses where the ping spread method of discovery is in use.
nodeDownDeleteInterval	Time interval	7d	The amount of time after an IP object ceases responding to ping before it is deleted.
doUnnumberedIp	Boolean	false	Switch to discover or ignore unnumbered IP networks.
numberOfRouteEntriesPoll	Integer	1000	Maximum number of routing table entries to be polled.
unmanageNonsnmp	Boolean	false	Switch to manage discovered IP objects which don't support snmp. When set to true and IP discovery discovers an IP object without an snmp agent, that object will not be monitored (pinged) by IP discovery. However, this will mean that if non-snmp systems are removed from the network, they will never be deleted.
manageAllNetworks	Boolean	true	Switch to manage all IP networks within scope or not. If set to false, IP discovery will only monitor networks around seed IP objects.
dumpFile	String	/dev/null	Dump filename. If not set, dumps will go to the console.
logFile	String	/dev/null	Log filename. If not set, the log will go to the console.
snmpFile	String	/dev/null	SNMP parameters filename. If not found, IP discovery will use defaults.

Keyword	Type	Default	Description
scope	IP Address Range	*.*.*.*	Defines the scope of IP addresses within which IP objects will be discovered. If scope is set to *.*.*.* (the default), all networks that can be will be discovered. However, the parameter, manageAllNetworks will be forced to false to monitor only IP networks around the seeded nodes.
limited	IP Address Range	null	Limits the IP address ranges within which IP objects will be discovered.
seeds	IP Addresses	null	IP addresses, separated by commas, from which IP discovery will begin discovery.
dhcp	IP Addresses	null	IP addresses, separated by commas, defining DHCP IP addresses.
debugMask	String	null	Specify parameters for IP discovery to output diagnostic information during execution. Parameters are separated by commas and include any of the following: snmp, ping, discovery and topology. For example, debugMask=snmp,discovery,topology.

D.6.2 Parameters for SNMP configuration file

The SNMP configuration file is used to specify SNMP configuration parameters. The file contains lines each of which is a sequence of SNMP configuration attributes. Blank lines are ignored and lines with a # in column 1 are treated as comments.

The format of a file is as follows:

```
ipRange:pingTimeout:pingRetry:statusPollInterval:getCommunityName:snmpTimeout:
snmpRetry:statusPollinterval:setCommunityName:remotePort:proxy
```

Table 30 summarizes the parameters that can be configured in this file.

Table 30. Parameter specifications for the IP discovery configuration file

Attribute	Type	Default	Description
ipRange	IP Address Range	n/a	The IP address or addresses for which the following attributes apply.
pingTimeout	Time interval	2s	The timeout interval applied to pings.
pingRetry	Integer	3	The maximum number of times ping will be attempted.
statusPollInterval	Time interval	1d	The time interval between pings of discovered resources.
getCommunityName	String	public	Community string to be used on SNMP GET requests. Multiple community strings can be specified separated by commas. Multiple GET requests will be attempted using each string in turn until the request is satisfied.
snmpTimeout	Time interval	2s	The timeout interval applied to SNMP requests.

Attribute	Type	Default	Description
snmpRetry	Integer	3	The maximum number of times an SNMP request will be attempted.
statusPollInterval	Time interval	1d	The time interval between SNMP polling of discovered resources.
setCommunityName	String	public	Community string to be used on SNMP SET requests. This is not used by IP discovery.
remotePort	Integer	161	The port used by remote SNMP agents.
proxy	IP Address or Hostname	*	The IP address or hostname of the proxy to be used, if required.

D.6.3 Running and troubleshooting

If the keyword parameter, logFile in the IP discovery configuration file is set to a valid HFS file name, various data produced by IP discovery will be written to that file. This includes debug information if the debugMask keyword parameter is configured.

An object will not be deleted from RODM until the period defined by the IP discovery configuration file keyword parameter, nodeDownDeleteInterval, has expired. By default, this is seven days. To override the default, set the parameter to a different value.

It may be useful to make the following changes to the OMVS initialization member (BPXPRMxx) in SYS1.PARMLIB:

- MAXTHREADS(10000). This prevents a problem where IP discovery ends due to an inability to create more (Java) threads.
- MAXASSIZE(2147483647). This prevents a problem where IP discovery ends due to a lack of memory available to individual address spaces running in UNIX System Services.

The above changes can be made dynamically using the `SETOMVS MVS` console command.

Ensure TRACE RESOLVER in the TCP/IP configuration is commented out (semicolon in column 1). Because of the nature of IP discovery, a lot of extraneous data is produced when this tracing is active which can cause critical load problems between UNIX Systems Services and NetView.

Related NetView Commands;

- `STOPDISC`: Stops the IP discovery sample.
- `STRTDISC`: Starts the IP discovery sample.

These commands can be used any time after an INITTOPO has been issued. Refer to the *Tivoli NetView for OS/390 Command Reference Volume 2 Version 1 Release 3*, SC31-8735 or online command help for INITTOPO keyword information.

D.7 Operational tasks

In this chapter we will describe all the necessary steps to start up our environment, then how to check its functionality and how to do an ordered shut down of all components.

D.7.1 Initialization operations

The following are the procedures we used to bring up all NetView components that are discussed in this book:

1. Verify that the TCP/IP stack and the SNMP agent are running on the OS/390 system.
2. Start the NetView Subsystem Interface with the following command from the system console:

   ```
   S NETVS03
   ```

   If the SSI starts correctly the following messages will appear:

```
S NETVS03
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 393
        NETVS03 WITH JOBNAME NETVS03. RACF WILL USE ICHRIN03.
$HASP100 NETVS03  ON STCINRDR
IEF695I START NETVS03  WITH JOBNAME NETVS03  IS ASSIGNED TO USER
++++++++
$HASP373 NETVS03  STARTED
IEF403I NETVS03 - STARTED - TIME=14.38.21
CNM226I NETVIEW PROGRAM TO PROGRAM INTERFACE
         INITIALIZATION IS COMPLETED
CNM541I NETVIEW SUBSYSTEM INITIALIZED SUCCESSFULLY
```

Figure 205. SSI startup messages

3. Start the RODM address space.

   There are two ways to start RODM:

   a. Cold start

   b. Warm start

Cold start

The cold start is used to bring up RODM with no structure and no objects loaded on it. If RODM is cold started, then the load job has to be submitted.

The following is the cold start procedure:

```
S EKGXRODM,TYPE=C,NAME=RODMNAME
```

and reply with `1` to the EKG1918D message that will appear.

If RODM starts up normally, these messages will appear:

```
S EKGXRODM,TYPE=C,NAME=RODMNAME
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 444
        EKGXRODM WITH JOBNAME EKGXRODM. RACF WILL USE ICHRIN03.
$HASP100 EKGXRODM ON STCINRDR
IEF695I START EKGXRODM WITH JOBNAME EKGXRODM IS ASSIGNED TO USER
++++++++
$HASP373 EKGXRODM STARTED
IEF403I EKGXRODM - STARTED - TIME=14.51.39
EKG1906I EKGXRODM : THE RODM NAME IS RODMNAME.
EKG1901I EKGXRODM : NO INIT METHOD IS SPECIFIED.
365 EKG1918D EKGXRODM : RODM RODMNAME WILL COLD
      START. ENTER '1' TO CONTINUE OR '2' TO TERMINATE.
R 365,1
IEE600I REPLY TO 365 IS;1
EKG1998I EKGXRODM : THE AUTHORITY CHECKING FOR THE RODM IS BYPASSED.
EKG0002I EKGXRODM : THE CURRENT ACTIVE LOG FILE IS NOW EKGLOGP.
IEC161I 227-229,EKGXRODM,EKGXRODM,EKGD003
EKG5011I EKGXRODM : THE NUMBER OF CHECKPOINT FILES USED BY RODM IS 2.
EKG1900I EKGXRODM : RODM RODMNAME INITIALIZATION IS COMPLETE WITH
LE/370.
```

Figure 206. RODM cold startup messages

Submit job INSTALL(CNMSJ012) to load the RODM with structure information. The following is the sample job that we used in our implementation:

```
//LOADDM   EXEC EKGLOADP,
//  RODMNAME=RODMNAME,     name of GMFHS's RODM (change or del. parm)
//  EKGIN1=NULLFILE,       EKGIN1 overridden - see below
//  EKGIN3=NULLFILE,       not needed for structure load
//  LOAD=STRUCTURE,
//  OPER=LOAD,
//  LISTL=ERRORSYNTAX,
//  SEVERITY=WARNING
//*
//LOADRODM.EKGIN1 DD DSN=NETVIEW.V1R3M0.CNMSAMP(DUIFSTRC),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM1),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM2),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM3),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM4),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM5),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM6),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM7),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM8),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDM9),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDMA),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDMB),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDMC),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDMD),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDME),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDMF),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLBTRDMZ),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLCSDM1),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLCSDM2),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLCSDM3),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLCSDM4),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLCSDM5),DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.CNMSAMP(FLCSDM8),DISP=SHR
```

Figure 207. Sample CNMSJ012 job

This job should finish with RC=0 and these are the messages that you can find in the syslog:

```
$HASP100 LUCY2    ON INTRDR      LUCY                  FROM TSU01461
LUCY
IRR010I  USERID LUCY     IS ASSIGNED TO THIS JOB.
ICH70001I LUCY     LAST ACCESS AT 09:45:13 ON TUESDAY, MARCH 14, 2000
$HASP373 LUCY2    STARTED - INIT B    - CLASS A - SYS RA03
IEF403I LUCY2 - STARTED - TIME=15.05.28
-                                                --TIMINGS (MINS.)--
        ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-LUCY2    LOADDM   LOADRODM    00   453   1426    .32    .00     .4
1464K   0     0     0     0     0
IEF404I LUCY2 - ENDED - TIME=15.05.53
-LUCY2    ENDED.  NAME-LUCY                TOTAL TCB CPU TIME=   .32
TOTAL ELAPSED TIME=    .4
$HASP395 LUCY2    ENDED
```

Figure 208. CNMSJ012 output

Warm start

Warm start is used only if the RODM has previously checkpointed on DASD with the command:

```
F EKGXRODM,CHKPT
```

This is the command output:

```
F EKGXRODM,CHKPT
EKG1302I EKGXRODM : RODM RODMNAME IS NOW CHECKPOINTING.
EKG1115I EKGXRODM : THE TRANSLATION WINDOW CHECKPOINT IS COMPLETE.
EKG1303I EKGXRODM : RODM RODMNAME HAS COMPLETED CHECKPOINTING.
```

Figure 209. RODM checkpoint

The command to start RODM is the following:

```
S EKGXRODM,NAME=RODMNAME
```

The command output is the following:

```
S EKGXRODM,NAME=RODMNAME
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 883
        EKGXRODM WITH JOBNAME EKGXRODM. RACF WILL USE ICHRIN03.
$HASP100 EKGXRODM ON STCINRDR
IEF695I START EKGXRODM WITH JOBNAME EKGXRODM IS ASSIGNED TO USER
++++++++
$HASP373 EKGXRODM STARTED
IEF403I EKGXRODM - STARTED - TIME=16.07.41
DUI4092E GMFHS UNABLE TO COMMUNICATE WITH RODM RODMNAME
EKG1905I EKGXRODM : A WARM START IS IN PROGRESS.
EKG1906I EKGXRODM : THE RODM NAME IS RODMNAME.
EKG1901I EKGXRODM : NO INIT METHOD IS SPECIFIED.
EKG1998I EKGXRODM : THE AUTHORITY CHECKING FOR THE RODM IS BYPASSED.
IEC161I 227-229,EKGXRODM,EKGXRODM,EKGD003
EKG5011I EKGXRODM : THE NUMBER OF CHECKPOINT FILES USED BY RODM IS 2.
EKG0002I EKGXRODM : THE CURRENT ACTIVE LOG FILE IS NOW EKGLOGP .
EKG1900I EKGXRODM : RODM RODMNAME INITIALIZATION IS COMPLETE WITH
         LE/370
```

Figure 210. RODM warm startup

4. Start the NetView program

   Issue the following command from the system console:

   ```
   S NETVA03
   ```

   Reply with an enter to the EZL603A message to start the AON functions. The messages produced by NetView startup are many, so we will show only the most important ones:

```
S NETVA03
$HASP100 NETVA03  ON STCINRDR
IEF695I START NETVA03  WITH JOBNAME NETVA03  IS ASSIGNED TO USER TCPIP3
 , GROUP OMVSGRP
$HASP373 NETVA03  STARTED
IEF403I NETVA03 - STARTED - TIME=15.09.34
BNJ080I BNJLINTB - BUFFER SIZE=24K,SLOT SIZE=200
BNH191I OPERATOR SECURITY SETTINGS: OPERSEC=NETVPW, OPSPAN=NETV
BNH193I COMMAND SECURITY SETTINGS: CMDAUTH=SCOPE, AUTHCHK=TARGETID,
TBLNAME=N/A, BACKTBL=N/A, SAFNODEC=N/A
BNH180I SPAN SECURITY SETTINGS: SPANAUTH=VTAMLST, SPANTBL=N/A,
SPANCHK=TARGETID, MVSSPAN=NO
BNH292I WEB BROWSER SECURITY SETTINGS: WEBAUTH=PASS, WEBSEC=PASS,
WEBIDLE=0 SECONDS
BNH350I NETVIEW HIGH-LEVEL LANGUAGE SUPPORT HAS BEEN SUCCESSFULLY
INITIALIZED WITH LE/370 LIBRARIES
*366 DSI802A RA03N    REPLY WITH VALID NCCF SYSTEM OPERATOR COMMAND
DSI410I DSIPARM MEMBER DSITBL01 BEING USED FOR NETVIEW AUTOMATION
CNM570I STARTING AUTOMATION TASK AUTO1
DSI530I 'AUTO1' : 'OST' IS READY AND WAITING FOR WORK
DSI166I DSILOG IS ACTIVATED BY RA03NPPT
DSI530I 'DSILOG  ' : 'DST' IS READY AND WAITING FOR WORK
DSI633I DEFAULTS COMMAND SUCCESSFULLY COMPLETED
CNM574I CONSOLE *MASTER* HAS BEEN ASSIGNED TO AUTO2
CNM570I STARTING AUTOMATION TASK AUTO2
DSI166I NTVASSIR IS ACTIVATED BY RA03NPPT
*367 EZL603A AUTOMATION - REPLY 'STOP', 'PAUSE', 'MSG=MEMBER',
  'CFG=CONTROL FILE' OR ENTER TO CONTINUE
REPLY 367 SUPPRESSED
IEE600I REPLY TO 367 IS;SUPPRESSED
DSI530I 'CNMCALRT':'ALERT RECEIVER TASK' IS READY AND WAITING FOR WORK
DWO624I MANAGEMENT SERVICES TRANSPORT INITIALIZED.
BNH020I DSI6SCF PROCESSED SUCCESSFULLY.  0 ENTRY POINTS WERE ADDED TO
THE SPHERE OF CONTROL
AAU136I SESSION STAT FUNCTION IS NOW ACTIVE
DUI390I COS GATEWAY COMMAND PROCESSOR SUCCESSFULLY INITIALIZED.
FLB447I SNA TOPOLOGY MANAGER IS INITIALIZING
DUI579I STATUS FOCAL POINT RA03N AND STATUS COLLECTOR RA03N ARE
SYNCHRONIZED.
DSI530I 'DUIFEAUT' : 'OST' IS READY AND WAITING FOR WORK
FLB678W SNA TOPOLOGY MANAGER FAILED TO CONNECT TO CMIP SERVICES AND
WILL RETRY, CMIP SERVICES IS NOT ACTIVE
FLB670I SNA TOPOLOGY MANAGER HAS ESTABLISHED COMMUNICATIONS WITH THE
RESOURCE STATUS FOCAL POINT
**** SETTING HARDWARE MONITOR FILTERS
CNM354I PDFILTER : COMMAND COMPLETE
DSI530I 'AUTODIS1' : 'OST' IS READY AND WAITING FOR WORK
```

Figure 211. NetView procedure startup

5. Start the GMFHS address space from the system console with the following command:

   ```
   S CNMGMFHS.GMFHS
   ```

   You will see the following messages on the system log:

```
S CNMGMFHS.GMFHS
IRR813I NO PROFILE WAS FOUND IN THE STARTED CLASS FOR 760
        CNMGMFHS WITH JOBNAME CNMGMFHS. RACF WILL USE ICHRIN03.
$HASP100 CNMGMFHS ON STCINRDR
IEF695I START CNMGMFHS WITH JOBNAME CNMGMFHS IS ASSIGNED TO USER
++++++++
$HASP373 CNMGMFHS STARTED
IEF403I CNMGMFHS - STARTED - TIME=15.24.35
DUI4027I GMFHS MAIN TASK INITIALIZATION IS COMPLETE FOR DOMAIN = RA03N
DUI4003I GMFHS NETWORK CONFIGURATION INITIALIZED SUCCESSFULLY
DUI3905I GMFHS NETWORK CONFIGURATION COMPLETED SUCCESSFULLY FOR DOMAIN
= RA03N
```

Figure 212. GMFHS startup messages

6. Log on to NetView.
7. Issue the following command to start the UNIX server:

   ```
   START UNIXSERV=*
   ```

   You will see the following messages on an NCCF screen:

```
NCCF                    Tivoli NetView   RA03N LUCY     03/14/00 16:28:21
* RA03N    START UNIXSERV=*
- RA03N    DSI360I START REQUEST IN PROGRESS
* RA03N    CNMESTSO CNMEUNIX *        *    CNMSJUNX LUCY     0
- RA03N    CNM279I CNMEUNIX(JOB01967) SUBMITTED
C RA03N    DSI633I START COMMAND SUCCESSFULLY COMPLETED
------------------------------------------------------------------------------




???
```

Figure 213. UNIX server startup messages

8. Start the Java-based Tivoli SNMP stack (JSNMP) with the following command:

   ```
   NETVASIS nvsrvc start
   ```

   Refer to Figure 204 on page 259 for the output messages.
9. Start the NMC topology server from a Windows NT desktop. Refer to 5.3.1, “Starting the NMC topology server” on page 149.
10. Start the communication between the NMC topology server and NetView with NETCONV command. Refer to 5.3.2, “Starting communication with host” on page 150.
11. Start the NMC topology console from Windows NT desktop. Refer to 5.3.3, “Starting the NMC topology console” on page 151.
12. Initialize the MultiSystem Manager by issuing the following command from an NCCF panel:

    ```
    INITTOPO
    ```

    The following messages will appear:

```
NCCF                    Tivoli NetView   RA03N LUCY     03/14/00 16:49:41
* RA03N     INITTOPO
C RA03N     FLC059I MULTISYSTEM MANAGER INITIALIZATION FILE FLCAINP HAS BEEN
            READ SUCCESSFULLY.  THE MULTISYSTEM MANAGER IS NOW ENABLED.
- RA03N     DSI268I EXCMD COMPLETE
--------------------------------------------------------------------------------




???
```

Figure 214. INITTOPO initialization messages

D.7.2 Operation verification

If everything has been started correctly you will be able to issue NVSNMP commands from a NetView console and the NMC will have the MultiSysView available as shown in this picture:

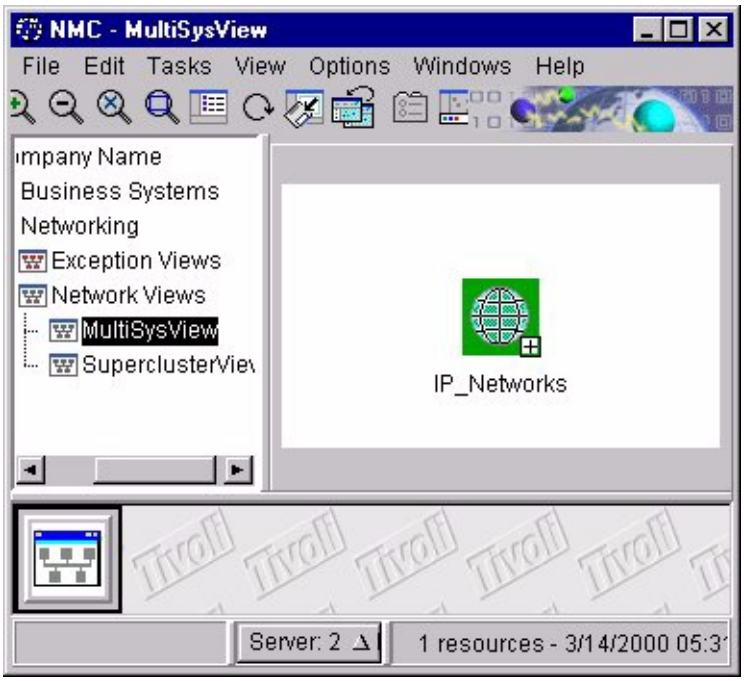


Figure 215. NMC and MultiSysView view

You can also issue the following two commands to check if everything is up and running. The first one is:

```
D A,L
```

The output will show you all the active address spaces on your system. Ensure that the highlighted ones are active:

```
  Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF OPERLOG  DATE 03/15/2000     4 WTORS                2 RESPONSES NOT SHOWN
RESPONSE=RA03
 IEE114I 09.23.02 2000.075 ACTIVITY 780
  JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
 00007    00020    00004      00028    00029    00004/00020       00025
  LLA      LLA      LLA      NSW  S  VLF      VLF      VLF      NSW  S
  APPC     APPC     APPC     NSW  S  ASCH     ASCH     ASCH     NSW  S
  JES2     JES2     IEFPROC  NSW  S  NET03    NET03    NET      NSW  S
  HSM03    HSM03    DFSMSHSM NSW  S  OSASF    OSASF    OSASF    IN   S
  SYSLOGD1 STEP1    OMVSKERN OWT  AO TSO      TSO      TCAS     OWT  S
  CICS03   CICS03   CICS     NSW  S  FTPD1    STEP1    TCPIP3   OWT  AO
  IHV      IHV      PSTEP01  OWT  S  RMF      RMF      IEFPROC  NSW  S
  TCPIPA   TCPIPA   TCPIP    NSW  SO OMPROUTA OMPROUTA *OMVSEX  IN   SO
  IOASRV   IOASRV   SERVER   OWT  SO OSNMPD   OSNMPD   OSNMPD   OWT  SO
  FTPDA1   STEP1    TCPIP3   OWT  AO NETVS03  NETVS03  NETVIEW  NSW  S
  EKGXRODM EKGXRODM START    NSW  S  NETVA03  NETVA03  NETVIEW  NSW  SO
  CNMEUNIX *OMVSEX           OWT  JO CNMGMFHS GMFHS    STEP1    NSW  S
  CNMEUNIX STEP1    TCPIP3   OWT  AO CNMEUNIX *OMVSEX  TCPIP3   IN   AO
COMMAND INPUT ===>                                          SCROLL ===> CSR
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
 F7=UP        F8=DOWN      F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 216. Active Subsystems display

The second one shows the active TCP/IP connections and checks if those highlighted are up. The command is:

```
D TCPIP,,N,CON
```

The output should be something like this:

```
RESPONSE=RA03
EZZ2500I NETSTAT CS V2R8 TCPIPA 782
USER ID  CONN     LOCAL SOCKET           FOREIGN SOCKET         STATE
BPXOINIT 00000024 0.0.0.0..10007         0.0.0.0..0             LISTEN
CNMEUNIX 0000291B 0.0.0.0..2099          0.0.0.0..0             LISTEN
CNMEUNIX 00002926 0.0.0.0..4099          0.0.0.0..0             LISTEN
CNMEUNIX 00002927 0.0.0.0..4546          0.0.0.0..0             LISTEN
CNMEUNIX 00002919 0.0.0.0..4545          0.0.0.0..0             LISTEN
FTPDA1   00000B39 0.0.0.0..21            0.0.0.0..0             LISTEN
IOASRV   00000015 0.0.0.0..8062          0.0.0.0..0             LISTEN
NETVA03  00002626 0.0.0.0..4020          0.0.0.0..0             LISTEN
NETVA03  0000262B 0.0.0.0..9998          0.0.0.0..0             LISTEN
NETVA03  00002629 0.0.0.0..9999         0.0.0.0..0            LISTEN
OMPROUTA 00000018 127.0.0.1..1026        127.0.0.1..1027        ESTABLS
OSNMPD   0000004B 0.0.0.0..1029          0.0.0.0..0             LISTEN
TCPIPA   00000029 0.0.0.0..9923          0.0.0.0..0             LISTEN
TCPIPA   00000017 127.0.0.1..1025        127.0.0.1..1028        ESTABLS
TCPIPA   00000025 0.0.0.0..23            0.0.0.0..0             LISTEN
TCPIPA   00000028 0.0.0.0..8823          0.0.0.0..0             LISTEN
TCPIPA   00000014 127.0.0.1..1027        127.0.0.1..1026        ESTABLS
TCPIPA   00000016 127.0.0.1..1028        127.0.0.1..1025        ESTABLS
TCPIPA   00000026 0.0.0.0..6623          0.0.0.0..0             LISTEN
TCPIPA   0000000D 0.0.0.0..1025          0.0.0.0..0             LISTEN
TCPIPA   00000027 0.0.0.0..7723          0.0.0.0..0             LISTEN
CNMEUNIX 00002928 0.0.0.0..7775          *..*                   UDP
OSNMPD   0000004A 0.0.0.0..161           *..*                   UDP
SYSLOGD1 0000002A 0.0.0.0..514           *..*                   UDP
24 OF 24 RECORDS DISPLAYED
```

Figure 217. Display of TCP/IP connections

Check that all the highlighted ports are active. This is the usage of each port:

2099 MIB and SNMP services (JSNMP)

4099 CLI services (JSNMP)

4020 NetView CNMTAMEL (task for graphic option)

9998 NetView DSIWBTSK (task for Web access)

9999 NetView TCPIP Task (task for TCP/IP workstation access)

7775 JSNMP service (trace purposes only)

D.7.3 Shutdown

The following are the required commands to perform an ordered shutdown of all the components used in this project:

1. Close the NMC by selecting File from the window's menu and then Exit.
2. Close the Topology Server and the Topology Communication Server by double-clicking the top-left corner of both the DOS boxes on your desktop. The following messages will appear on the NetView screen:

```
NCCF                    Tivoli NetView   RA03N LUCY     03/14/00 17:35:38
- RA03N    DUI419I COMMUNICATION TO IP 23-8P4YK.itso.ral.ibm.com (9.24.106.77)
           TERMINATED NORMALLY.  THE COMMUNICATION SERVER CLOSED THE SOCKET.
------------------------------------------------------------------------------




???
```

Figure 218. Topology Server shut down messages

3. Close the Java-based Tivoli SNMP stack (JSNMP) using the following command on a NetView console:

   ```
   netvasis nvsrvc end
   ```

   You will see the following messages on your screen:

```
NCCF                    Tivoli NetView   RA03N LUCY     03/14/00 17:38:31
* RA03N    nvsrvc end
x RA03N    DSI037I Process 50331912 spawned for 'nvsrvc  end'.
x RA03N    nvsrvc stop....
x RA03N    +0000000000 COMMAND nvsrvc  end
------------------------------------------------------------------------------




???
```

Figure 219. JSNMP shutdown messages

4. Close the UNIX Command Server with the following command on the NetView console:

   ```
   STOP UNIXSERV=*
   ```

   You will see the following messages:

```
NCCF                    Tivoli NetView   RA03N LUCY     03/14/00 17:40:58
* RA03N    STOP UNIXSERV=*
- RA03N    DSI056I UNIX SESSION STOPPING FOR *
------------------------------------------------------------------------------




???
```

Figure 220. UNIX command server shutdown messages

5. Close the NetView procedure by issuing the following command from a NetView console:

   ```
   CLOSE IMMED
   ```

   The following messages will appear on the system log:

```
IEA631I  OPERATOR NTVASSIR NOW INACTIVE, SYSTEM=RA03    , LU=NTVASSIR
IEA631I  OPERATOR AUTO1    NOW INACTIVE, SYSTEM=RA03    , LU=AUTO1
DUI383I DUIFSSCO SCOPE CHECKER OPT TERMINATION IN PROGRESS.
DUI381I DUIFSSCO SCOPE CHECKER OPT HAS TERMINATED.
DSI200I TASK BNJMNPDA HAS TERMINATED
DSI200I TASK EZLTDDF HAS TERMINATED
DSI531I 'CNMCALRT' : 'ALERT RECEIVER TASK' IS TERMINATING
FLB444E SNA TOPOLOGY MANAGER ENCOUNTERED A PROCESSING ERROR
FLB442E SNA TOPOLOGY MANAGER IS SHUTTING DOWN BECAUSE OF AN ERROR
DSI200I TASK DSI6DST HAS TERMINATED
DSI200I TASK DSIELTSK HAS TERMINATED
DSI200I TASK DSIGDS HAS TERMINATED
DSI200I TASK DUIFSSCO HAS TERMINATED
DSI200I TASK DSIHPDST HAS TERMINATED
IST593I  ISTPDCLU SESSION AWARENESS SESSION ENDED
IST593I  ISTPDCLU PD TRACE          SESSION ENDED
DSI200I TASK DSIATOPT HAS TERMINATED
DSI200I TASK DSIQTSK HAS TERMINATED
FLB610I TASK FLBTOPO IS STARTING LOGOFF PROCESSING
FLB611I TASK FLBTOPO HAS COMPLETED ITS LOGOFF PROCESSING
DSI200I TASK CNMCALRT HAS TERMINATED
DSI200I TASK EZLTCFG HAS TERMINATED
DSI200I TASK RA03NVMT HAS TERMINATED
DSI200I TASK DSIROVS HAS TERMINATED
DSI200I TASK DSIUDST HAS TERMINATED
..........
DSI200I TASK DSIMONIT HAS TERMINATED
DSI111I NCCF ABNORMALLY TERMINATING - CLOSE IMMED REQUESTED BY LUCY
DSI135I NCCF TERMINATION COMPLETE
-                                                  --TIMINGS (MINS.)--
        ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-NETVA03           NETVIEW     00  12406  33563   1.14    .02  155.3
36239K   0     0     0     0     0
IEF404I NETVA03 - ENDED - TIME=17.44.55
-NETVA03  ENDED.  NAME-                    TOTAL TCB CPU TIME=  1.14
TOTAL ELAPSED TIME= 155.3
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 NETVA03  ENDED
```

Figure 221. NetView procedure shutdown

6. Close the NetView subsystem interface with the following command on the system console:

   ```
   P NETVS03
   ```

   You will see the following messages on the system log:

```
P NETVS03
CNM632I NETVIEW PROGRAM TO PROGRAM INTERFACE IS BEING TERMINATED
CNM580I NETVIEW SUBSYSTEM ADDRESS SPACE IS TERMINATING
DUI4202E TEMPORARY PPI FAILURE, RETURN CODE = 0, ECB POST CODE = 99
IST1205I MANAGEMENT SERVICES TRANSPORT UNAVAILABLE 372
IST1208I PROGRAM-TO-PROGRAM INTERFACE MODULE IS INACTIVE
IST314I END
-                                                  --TIMINGS (MINS.)--
        ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-NETVS03           NETVIEW     00     52    328    .00    .00  190.9
1834   0     0     0     0     0
IEF404I NETVS03 - ENDED - TIME=17.49.21
-NETVS03  ENDED.  NAME-                     TOTAL TCB CPU TIME=   .00
TOTAL ELAPSED TIME= 190.9
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 NETVS03  ENDED
```

Figure 222. NetView subsystem interface shutdown

7. Close the GMFHS address space with the following command from the system console:

   ```
   P GMFHS
   ```

   The following messages will appear on the system console:

```
P GMFHS
DUI4031I GMFHS IS TERMINATING OR IS IN THE PROCESS OF TERMINATING DUE
TO OPERATOR REQUEST FOR DOMAIN = RA03N
-                                                  --TIMINGS (MINS.)--
        ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-CNMGMFHS          STEP1       00    154    612    .13    .04  147.8
920K   0     0     0     0     0
IEF404I CNMGMFHS - ENDED - TIME=17.52.23
-CNMGMFHS ENDED.  NAME-                     TOTAL TCB CPU TIME=   .13
TOTAL ELAPSED TIME= 147.8
$HASP395 CNMGMFHS ENDED
```

Figure 223. GMFHS shutdown messages

8. Close the RODM address space with the following command on the system console:

   ```
   F EKGXRODM,TERM
   ```

   The following messages will appear on the system console:

```
F EKGXRODM,TERM
EKG1916I EKGXRODM : RODM RODMNAME TERMINATION IS IN PROGRESS.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=EKGXRODM, ASID=005D.
EKG1917I EKGXRODM : RODM RODMNAME TERMINATION IS COMPLETE.
-                                                  --TIMINGS (MINS.)--
        ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
SERV  PG  PAGE  SWAP   VIO SWAPS
-EKGXRODM          START       00    612   3559    .01    .00  184.5
191K   0     0     0     0     0
IEF404I EKGXRODM - ENDED - TIME=17.56.14
-EKGXRODM ENDED.  NAME-                     TOTAL TCB CPU TIME=   .01
TOTAL ELAPSED TIME= 184.5
IEF352I ADDRESS SPACE UNAVAILABLE
$HASP395 EKGXRODM ENDED
```

Figure 224. RODM shutdown messages

Appendix E. Configuration files used at ITSO Raleigh Center

This appendix includes some configuration files that were used for the OS/390 systems in the ITSO Raleigh test installation of CS for OS/390 V2R8 IP.

E.1 PROFILE.TCPIP on RA39

```
;****************************** Top of Data ***********************
; Member TCPIP.TCPPARMS(PROF39A)
; ********************************************************************
;
DATASETPREFIX TCPIP
;
ITRACE OFF
; ITRACE ON SUBAGENT 3
;
TCPCONFIG
;  INTerval  5             ; In minutes - Keep alive packet 0-35791
;  RESTRICTLowports
   UNRESTRICTLowports
   TCPSENDBfrsize  16384   ; Range is 256-256K - Default is 16K
   TCPRCVBufrsize  16384   ; Range is 256-256K - Default is 16K
   SENDGARBAGE FALSE       ; Packet contains no data
;  SENDGARBAGE TRUE        ; Packet contains 1 byte of random data

UDPCONFIG
;  RESTRICTLowports
   UNRESTRICTLowports
   UDPCHKsum               ; Do checksum
;  NOUDPCHKsum             ; Don't do checksum
   UDPSENDBfrsize 16384    ; Range is ???-???K (Default is 65536 bytes)
   UDPRCVBufrsize  16384   ; Range is ???-???K (Default is 65536 bytes)
;  UDPQueuelimit           ; Limit inbound UDP Queue ????
;  NOUDPQueuelimit         ; Do not Limit inbound UDP Queue ????
;
; ********************************************************
IPCONFig
 ARPTO  1200           ; In seconds
;CLAWUSEDoublenop      ; Applies only to first-level MVS systems
 DATAGRamfwd
;NODATAGRamfwd
;FIREWALL
 NOSOURCEVIPA
;SOURCEVIPA
;NOVARSUBNETTING       ; For RIPV1
 VARSUBNETTING          ; For RIPV2
;NOSYSPLEXRouting
 SYSPLEXRouting
 IGNORERedirect
 REASSEMBLytimeout   15 ; In seconds
 STOPONclawerror
 TTL    60             ; In seconds, but actually Hop count
 MULTIPATH
; NODYNAMICXCF
 DYNAMICXCF 172.16.233.39 255.255.255.0 1
; ---------------------------------------------------------------------
```

```
SACONFIG  COMMUNITY MVSsubagent
 ENABLED
 AGENT 161
 SETSENABLED
 OSASF 760
 ATMENABLED
; SETSDISABLED
; ---------------------------------------------------------------------
;
;
; AUTOLOG the following servers.
;

 AUTOLOG 1
;   NAMED   JOBNAME NAMED1     ; Domain Name Server
;   FTPD JOBNAME FTPD1   ; FTP Server
    OMPROUTA             ; OSPF   Server
    IOASNMP              ; SNMP OSA-2 ATM Management
;   WEBSRV               ; DOMINO
;   OSNMPD               ; SNMP agent
;   SYSLOGD JOBNAME SYSLOGD1 ; SYSLOG daemon
;   GOWLMRG1             ; sysplex test server
 ENDAUTOLOG

;
; ---------------------------------------------------------------------
PORT
    20 TCP OMVS      NOAUTOLOG ; FTP Server
    21 TCP FTPD1               ; FTP Server
    21 TCP FTPDA1              ; FTP Server
;   23 TCP INTCLIEN            ; Telnet Server
    25 TCP omvs                ; SMTP Server
    53 TCP omvs                ; Domain Name Server - Parent Process
    53 UDP omvs                ; Domain Name Server - Parent Process
    80 TCP WEBSRV              ; Domino webserver
   135 UDP LLBD                ; NCS Location Broker
   161 UDP OSNMPD              ; SNMP Agent
   162 UDP OMVS                ; osnmp command (SNMP Manager)
 16200 UDP OMVS                ; osnmp command (SNMP Manager)
   443 TCP OMVS                ; Domino webserver
   512 TCP RSHDA               ; Remote Execution Server
   514 TCP RSHDA               ; Remote Execution Server
   514 UDP OMVS                ; SYSLOG daemon
   515 TCP T03ALPD             ; LPD Server
   580 UDP NCPROUT             ; NCPROUTE Server
;  760 UDP IOASNMP             ; osa/sf
   760 TCP IOASNMP             ; osa/sf


;
; ---------------------------------------------------------------------
; *******************************************************
; VIPA Definition  (For V2R5)
; *******************************************************
;
   DEVICE VIPA39A  VIRTUAL     0
   LINK   VIPA39A  VIRTUAL     0     VIPA39A
```

```
; ********************************************************
; LCS Definition
; osa ch D8                          Device # 2060-2061
; ********************************************************
 DEVICE TR1    LCS  2060 autorestart
 LINK  TR1    IBMTR    0 TR1
;
; ********************************************************
; LCS Definition
; OSA CHPID D8                        Device # 2064-2065
; ********************************************************
  DEVICE EN1  LCS           2064
  LINK   EN1  ETHEROR802.3  1     EN1
; ********************************************************
;  LINK   to 2216 mae
; ********************************************************
 DEVICE M392216B MPCPTP AUTORESTART
 LINK   M392216B MPCPTP M392216B
; ********************************************************
;  LINK   to 2216 400
; ********************************************************
 DEVICE M392216C MPCPTP AUTORESTART
 LINK   M392216C MPCPTP M392216C
;
; ********************************************************
; ATM OSA Definition for SNMP management
;     TRLE : USER.VTAMLST(RA39ATM1)
;          : USER.VTAMLST(RA39ATM2)
; ********************************************************
  DEVICE OSARA39A ATM   PORTNAME PORTE0
  LINK   OSARA39A ATM   OSARA39A
; ********************************************************
; ATM OSA Definition LE
; ********************************************************
  DEVICE OSA20A0 LCS   20A0
  LINK OSAT20A0  IBMTR 0 OSA20A0

  DEVICE OSA20A2 LCS   20A2
  LINK OSAT20A2  ETHEROr802.3 1 OSA20A2
;
; ---------------------------------------------------------------------
HOME
    172.16.232.39 VIPA39A     ; VIPA
    9.24.104.149   TR1        ; OSA2 TRN to public network
    9.24.105.73    EN1        ; Ethernet
    192.168.140.12 OSAT20A0
    192.168.140.75 OSAT20A2
    172.16.102.39 M392216B    ; MPC TO 2216 mae
;
; ---------------------------------------------------------------------
;
;  Dynamic VIPA Definitions
;
VIPADynamic
   VIPABackup   200  172.16.240.3
   VIPABackup   200  172.16.240.28
;  VIPABackup   30   172.16.240.40
   VIPADEFine   255.255.255.192 172.16.240.39
```

```
   VIPARange    DEFINE  255.255.255.192  172.16.240.192
;  VIPARange    DEFINE  255.255.255.192  172.16.240.200
ENDVIPADYNAMIC
;
GLOBALCONFIG
   TCPIPSTATISTICS
; ------------------------------------------------------------------------
 INCLUDE TCPIP.TCPPARMS.R2607(TELN&SYSCLONE.A)
; ------------------------------------------------------------------------
 START TR1                ; OSA
 START EN1                ; Ethernet
 START OSA20A0
 START OSA20A2
 START M392216b           ; mpc to 2216 mae
```

E.2 TCPIP.DATA on RA39

```
;*********************************************************************
;                                                                    *
;   Name of Data Set:     TCPIP.TCPPARMS(TCPD39A)                    *
;                                                                    *
;*********************************************************************
TCPIPJOBNAME TCPIPA
HOSTNAME  MVS39A

DOMAINORIGIN  itso.ral.ibm.com
NSINTERADDR  9.24.106.15
NSPORTADDR 53
;TRACE RESOLVER
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
DATASETPREFIX TCPIP
MESSAGECASE MIXED
; MESSAGECASE UPPER
; LOADDBCSTABLES SJISKANJI EUCKANJI
```

E.3 /etc/snmpd.conf

```
#
# /etc/snmpd.conf : CS for OS/390 V2R8 IP on SYSTEM 39
#
# Licensed Materials - Property of IBM
# "Restricted Materials of IBM"
# 5647-A01
# (C) Copyright IBM Corp. 1997, 1998
#
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
# Notes
# - All values for an entry must be on the same line.
# - All keys need to be regenerated using the pwtokey command in order
#   for these sample entries to actually be used.
# - In this sample:
#   - Keys are generated for use with engineID 0000000200000000009186895
#   - Authentication keys were generated with password of
#     "authpass" and are localized keys.
#     Note: The password could be different for every user              owed.
#   - Privacy keys generated with a password of "privpass".
#   - Entries defined to use encryption support, which is available only
#     as a separately orderable feature on the base OS/390 product, are
#     included below but commented out.
```

```
#--------------------------------------------------------------------------------------------
------------------------------------------------------------

#--------------------------------------------------------------------------------------------
------------------------------------------------------------
# USM_USER entries
# Format is:
#  userName engineID authProto authKey privProto privKey keyType storageType
#
#--------------------------------------------------------------------------------------------
------------------------------------------------------------
USM_USER u03 - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  -   -  L -

#USM_USER u03 - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97 DES
61a5dfe8a7277f1f982edce11e9fc633 L -

USM_USER  u03n - none -   -   -  L -

USM_USER u28 0000000200000000009186895 HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97 DES
da17ff9cb12620fe00aa5fed515b41da L nonVolatile

USM_USER  u28n - none -   -   -  L -

USM_USER  u39 - HMAC-MD5 b186f7ab80c9e7f8cda3dbc6f6203f97  -   -  L -

#USM_USER  u28a - HMAC-MD5 4a93130df02117a66dd80cbf1cf0ee9c  -   -  L -

#--------------------------------------------------------------------------------------------
------------------------------------------------------------
# VACM_GROUP entries
# Format is:
#  groupName securityModel securityName storageType
#--------------------------------------------------------------------------------------------
------------------------------------------------------------
#VACM_GROUP groupu03 USM u03 nonVolatile
VACM_GROUP group1 USM     u03  -
VACM_GROUP group1 USM     u03n -
VACM_GROUP group1 USM     u28  -
VACM_GROUP group1 USM     u28n -
VACM_GROUP group1 USM     u39  -

#--------------------------------------------------------------------------------------------
------------------------------------------------------------
VACM_GROUP group3 SNMPv1  publicv1  -
VACM_GROUP group3 SNMPv2c publicv2c -

VACM_GROUP group4 SNMPv1  MVSsubagent  -
VACM_GROUP group4 SNMPv2c MVSsubagent  -

#--------------------------------------------------------------------------------------------
------------------------------------------------------------
# VACM_VIEW entries
# Format is:
#  viewName viewSubtree viewMask viewType storageType
#--------------------------------------------------------------------------------------------
------------------------------------------------------------
VACM_VIEW bigView        internet   - included -

VACM_VIEW prettyBigView  internet   - included -
VACM_VIEW prettyBigView  interfaces - excluded -

VACM_VIEW mediumView     system     - included -
VACM_VIEW mediumView     interfaces - included -
VACM_VIEW mediumView     tcp        - included -
VACM_VIEW mediumView     udp        - included -
VACM_VIEW mediumView     icmp       - included -

VACM_VIEW smallView      snmp       - included -
VACM_VIEW v1View         internet   - included -
VACM_VIEW v1View         system     - excluded -

VACM_VIEW subagentView   dpiPort    - included -

#--------------------------------------------------------------------------------------------
------------------------------------------------------------
# VACM_ACCESS entries
# Format is:
#            group context context security security  read     write        notify  storage
```

```
#             Name  Prefix  Match   Level    Model    View     View         View     Type
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
VACM_ACCESS  group1  -       -      AuthPriv   USM     bigView   prettyBigView bigView   -
VACM_ACCESS  group1  -       -      AuthNoPriv USM     bigView   prettyBigView bigView   -
VACM_ACCESS  group1  -       -    noAuthNoPriv USM     bigView   smallView     bigView   -

#VACM_ACCESS  group3  -      -    noAuthNoPriv SNMPv1  smallView bigView       bigView   -
#VACM_ACCESS  group3 -       -    noAuthNoPriv SNMPv1  bigView   bigView       bigView   -
VACM_ACCESS  group3  -       -    noAuthNoPriv SNMPv1  v1View    bigView       bigView   -
VACM_ACCESS  group3  -       -    noAuthNoPriv SNMPv2c bigView   bigView       bigView   -

VACM_ACCESS  group4  -       -    noAuthNoPriv SNMPv1  subagentView   -             -    -
VACM_ACCESS  group4  -       -    noAuthNoPriv SNMPv2c subagentView   -             -    -

#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
# NOTIFY entries
# Format is:
#  notifyName tag type storageType
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
NOTIFY notify1 traptag trap -

#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
# TARGET_ADDRESS
# Format is:
#           target  tDomain tAddress          tagList  target  timeout retry  storage
#           AddrName                                   Params          Count   Type
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------

TARGET_ADDRESS Target1 UDP 9.24.104.113:16200 traptag trapparms1  -   -        -
TARGET_ADDRESS Target2 UDP 9.24.104.113:16200 traptag trapparms2  -   -        -
TARGET_ADDRESS Target3 UDP 9.24.104.113:16200  traptag trapparms3  -   -       -

#TARGET_ADDRESS Target4 UDP 127.0.0.1:16200    traptag trapparms1  -   -       -
#TARGET_ADDRESS Target5 UDP 127.0.0.1:16200    traptag trapparms2  -   -       -
TARGET_ADDRESS Target6 UDP 127.0.0.1:16200     traptag trapparms5  -   -       -

TARGET_ADDRESS Target7 UDP 9.24.104.42:16200   traptag trapparms4  -   -       -
TARGET_ADDRESS Target8 UDP 9.24.104.42:162     traptag trapparms1  -   -       -

#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
# TARGET_PARAMETERS
# Format is:
#                 params    mpModel security security   security   storage
#                  Name              Model    Name       Level      Type
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
TARGET_PARAMETERS trapparms1 SNMPv1  SNMPv1  publicv1  noAuthNoPriv  -
TARGET_PARAMETERS trapparms2 SNMPv2c SNMPv2c publicv2c noAuthNoPriv  -
TARGET_PARAMETERS trapparms3 SNMPv3  USM     u03       AuthNoPriv    -
#TARGET_PARAMETERS trapparms4 SNMPv3  USM     u28       AuthNoPriv    -
TARGET_PARAMETERS trapparms4 SNMPv3  USM     u28       AuthPriv    -
TARGET_PARAMETERS trapparms5 SNMPv3  USM     u39       AuthNoPriv    -

#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
# COMMUNITY
# Format is:
#        community    security    security       netAddr    netMask  storageType
#         Name         Name         Level
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
COMMUNITY publicv1    publicv1     noAuthNoPriv  0.0.0.0     0.0.0.0      -
COMMUNITY publicv2c   publicv2c    noAuthNoPriv  0.0.0.0     0.0.0.0      -
COMMUNITY MVSsubagent MVSsubagent  noAuthNoPriv  0.0.0.0     0.0.0.0      -

#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
# DEFAULT_SECURITY
# Format is:
#  securityPosture password privacy
#-------------------------------------------------------------------------------------------
-------------------------------------------------------------
```

```
DEFAULT_SECURITY semi-secure defaultpassword no

#--------------------------------------------------------------------
#  Any SNMP agent configuration entries added by dynamic configuration
# (SET) requests get added to the end of the SNMPD.CONF file.
#--------------------------------------------------------------------
```

E.4 /etc/osnmp.conf

```
#-----------------------------------------------
#-----------------------------------------------
# Community-based security  (SNMPv1 and SNMPv2c)
#-----------------------------------------------
v1       127.0.0.1   snmpv1
v2c      127.0.0.1   snmpv2c
mvs03a1  9.24.104.113 snmpv1
mvs03a   9.24.104.113 snmpv2c
mvs39a1  9.24.104.149 snmpv1
mvs39a   9.24.104.149 snmpv2c
#-------------------------------------------------------------
# User-based Security Model (USM with SNMPV3)
#
#-----------------------------------------------
#v3m39a  9.24.104.149 snmpv3 u03 password - AuthNoPriv  HMAC-MD5 -
-   -
#3m39a  9.24.104.149 snmpv3 u03 -    - AuthPriv   HMAC-MD5 9f1626506dbd7540c8ce526070fb69f9
DES 9f1626506dbd7540c8ce526070fb69f9
#
v3m03a  9.24.104.113 snmpv3 u03 -    - AuthPriv   HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3

v3m03a1 9.24.104.113 snmpv3 u03 -    - AuthNoPriv  HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
-   -

v3m03aE 9.24.104.113 snmpv3 u03 -    - AuthPriv   HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3

v3m03aE 172.16.250.3 snmpv3 u03 -    - AuthPriv   HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3

v3m03aF 9.24.104.113 snmpv3 u03 -    - AuthPriv   HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3

#3m03aF 172.16.250.3 snmpv3 u03 -    - AuthPriv   HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3
#
v3m28a  9.24.104.42  snmpv3 u03 -    - AuthNoPriv  HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
-   -

v3m28aN 9.24.104.42   snmpv3 u03n -    - noAuthNoPriv  - - -   -

v3m39a  9.24.104.149 snmpv3 u03 -    - AuthPriv   HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
DES 7865f0c012464308a5a4deead83903c3

v3m39a1 9.24.104.149 snmpv3 u03 -    - AuthNoPriv  HMAC-MD5 eb81360f73adf8d6967eaadec03dff45
-   -

v3m39aN 9.24.104.149 snmpv3 u03n -     - noAuthNoPriv  - -  -  -

v3m39a2 9.24.104.149 snmpv3 u04 -  - AuthNoPriv  HMAC-MD5 ba226e9841527101779f62ca75a8bbf9
-  -
```

E.5 OMPROUTE OSPF configuration file on RA39

```
Area    Area_Number=0.0.0.0
             Stub_Area=NO
             Authentication_type=None;
RouterID=172.16.232.39;
ROUTESA_CONFIG ENABLED=YES
            COMMUNITY="MVSsubagent";
OSPF_Interface IP_Address=172.16.102.39
```

```
                Name=m392216b
                Cost0=10
                Subnet_mask=255.255.255.0
                MTU=32768;
OSPF_Interface IP_Address=172.16.233.*
                Cost0=10
                Non_Broadcast=Yes
                Subnet_mask=255.255.255.0
                MTU=32768;
OSPF_Interface IP_Address=172.16.240.*
                name=DVIP39A
                Non_Broadcast=Yes
                Cost0=1
                Subnet_mask=255.255.255.192;
OSPF_Interface IP_Address=172.16.232.39
                name=VIPA39A
                Subnet_mask=255.255.255.0
                Non_Broadcast=Yes
                Destination_Addr=172.16.232.39
                Cost0=1
                MTU=32768;
OSPF_Interface IP_Address=9.24.104.149
                Name=TR1
                Cost0=60
                Subnet_mask=255.255.255.0
                MTU=4082;
OSPF_Interface IP_Address=9.24.105.73
                Name=EN1
                Cost0=66
                Subnet_mask=255.255.255.0
                MTU=1492;
OSPF_Interface IP_Address=192.168.140.12
                Name=OSAT20A0
                COST0=20
                Subnet_mask=255.255.255.240
                MTU=1492;
OSPF_Interface IP_Address=192.168.140.75
                Name=OSAT20A2
                COST0=20
                Subnet_mask=255.255.255.240
                MTU=1492;
```

E.6 ATM OSA-2 adapter configuration (CHPID=E0)

```
/*********************************************************************/
/*  OSA/SF Query created 09:52:55 on 03/07/2000                      */
/*  IOACMD APAR level - OW39984                                      */
/*********************************************************************/

*******************************
* CHPID E0 information follows *
*******************************
CHPID -------------------------> E0
Hardware model ----------------> OSA-2(ATM)
Subtype -----------------------> Logical Token Ring and Ethernet ports
Reset indicator ---------------> Reset complete
Modes configured --------------> TCP/IP Passthru
```

```
                                ATM (LE)
Control unit number ------------> 20A0
Processor code level -----------> 05.A4
Flash image level --------------> OSAATM NULL
ATM flash image level ----------> A:05.A2.00

This OSA/SF LP number(name) ----> 04 (A4)
Managing OSA/SF LP number(name)-> 04 (A4)
Channel path status ------------> Online
Shared -------------------------> Yes

Channel node descriptor --------> Valid
  Node -------------------------> device-type
  Class ------------------------> Communications controller
  Type and model ---------------> 009676-003
  Manufacturer and plant -------> IBM-42
  Sequence number and tag ------> SN537028133 0000

*******************************************
* OSA-2 CHPID E0 is defined to 8 partitions *
*******************************************
  LP (LP name)

  01 (A1)
  02 (A2)
  03 (A3)
  04 (A4)
  05 (A5)
  08 (A8)
  09 (A9)
  0A (A10)

*********************************************
* Port information follows for OSA-2 CHPID E0 *
*********************************************

*****************************************************************************
* Information for OSA-2 CHPID E0 physical port 0                            *
* Settable port parameters (using SET_PARM) are preceded by 's-'            *
* Configurable port parameters (using CONFIG_OSA) are preceded by 'c-' *
* Settable and configurable port parameters are preceded by 'p-'            *
*****************************************************************************
s-Hardware state -----------------> Enabled

  Transmission type --------------> 155 Mbps
  Media type ---------------------> MultiMode fiber
  UNI type -----------------------> Private
  Highest UNI version supported --> Version 3.1
  Active UNI version -------------> Version 3.1
  Received ATM cells -------------> 000000000000077B
  Transmitted ATM cells ----------> 0000000000000318
  Network prefix port ------------> 0
  ATM address network prefix -----> 399999999999990000011110221
  Network prefix status ----------> Valid
  Maximum VPCs supported ---------> 0
  Maximum VCCs supported ---------> 1024
  Number of active VPCs ----------> 0
  Number of active VCCs ----------> 13
```

```
c-Maximum number of VPI bits -----> 0
c-Maximum number of VCI bits -----> 11
  ILMI VPI ----------------------> 0
  ILMI VCI ----------------------> 16
  ATM address type ---------------> Private
  ATM physical address ----------->
                    39999999999999000011110221000 4AC20E695F0
  Non-error ATM cells discarded
     inbound ---------------------> 0
  HEC error ATM cells discarded
     inbound ---------------------> 0
  Header validation ATM cells
     discarded inbound -----------> 4
  Maximum AAL5 PDU size ----------> 4544
  Received AAL5 PDUs -------------> 799
  Non-error AAL5 PDU discards
     inbound ---------------------> 0
  Errored AAL5 PDU discards
     inbound ---------------------> 0
  Transmitted AAL5 PDUs ----------> 346
  Non-error AAL5 PDU discards
     outbound --------------------> 0
  ATM code operational status ----> Operational
  Universal MAC address ----------> 0004AC20E695
s-Active MAC address -------------> 0004AC20E695
  Max PCM connections ------------> 0
c-Port name ----------------------> PORTE0
s-Port description ---------------> ATM LANE TR
s-TCP/IP instance IP address -----> 0.0.0.0
  Object ID ----------------------> 1.3.6.1.4.1.1.3.12
c-ILMI ---------------------------> Enabled
c-SVCs ---------------------------> Enabled
c-Transmit clock source ----------> OSA
c-Physical layer type ------------> SONET


************************************************************************
* Information for OSA-2 CHPID E0 emulated port 0                       *
* Settable port parameters (using SET_PARM) are preceded by 's-'       *
* Configurable port parameters (using CONFIG_OSA) are preceded by 'c-' *
* Settable and configurable port parameters are preceded by 'p-'       *
************************************************************************
  Port type ----------------------> Token Ring (ATM LAN emulation)
s-Group address 1 ----------------> not set
s-Group address 2 ----------------> not set
s-Group address 3 ----------------> not set
s-Group address 4 ----------------> not set
s-Group address 5 ----------------> not set
s-Group address 6 ----------------> not set
s-Group address 7 ----------------> not set
s-Group address 8 ----------------> not set
s-Group address 9 ----------------> not set
s-Group address 10 ---------------> not set
s-Group address 11 ---------------> not set
s-Group address 12 ---------------> not set
s-Group address 13 ---------------> not set
s-Group address 14 ---------------> not set
s-Group address 15 ---------------> not set
```

```
s-Group address 16 ----------------> not set
s-Group address 17 ----------------> not set
s-Group address 18 ----------------> not set
s-Group address 19 ----------------> not set
s-Group address 20 ----------------> not set
s-Group address 21 ----------------> not set
s-Group address 22 ----------------> not set
s-Group address 23 ----------------> not set
s-Group address 24 ----------------> not set
s-Group address 25 ----------------> not set
s-Group address 26 ----------------> not set
s-Group address 27 ----------------> not set
s-Group address 28 ----------------> not set
s-Group address 29 ----------------> not set
s-Group address 30 ----------------> not set
s-Group address 31 ----------------> not set
s-Group address 32 ----------------> not set

  Multicast information

  LP number      Unit address         Multicast          Group
  (LP name)      (device number)      IP address         address

  4 (A4      )  00(20A0),01(20A1)   224.000.000.001    C00000040000
  4 (A4      )  00(20A0),01(20A1)   224.000.000.005    C00000040000
  4 (A4      )  00(20A0),01(20A1)   224.000.000.006    C00000040000


s-User data ----------------------->
  LE PDU octets inbound ----------> 033376
  Non-error LE PDU discards
     inbound ---------------------> 131
  Errored LE PDU discards inbound-> 0
  Unknown protocol LE PDU
     discards inbound ------------> 0
  LE PDU octets outbound ---------> 02920
  Non-error LE PDU discards
     outbound --------------------> 0
  Errored LE PDU discards
     outbound --------------------> 0
  Configuration mode -------------> Automatic
c-Configured LAN type ------------> Token ring
c-Configured max data frame size--> 4544
c-Configured ELAN name -----------> TKR_1
c-Configured LE server ATM
     address ---------------------> 00000000000000000000000000000000000000000000
c-Control timeout ----------------> 120
c-Max unknown frame count --------> 5
  Max unknown frame time (seconds)> 1
c-VCC timeout period (seconds) ---> 1200
c-Max retry count ----------------> 1
c-LE_ARP cache aging time --------> 300
c-Forward delay time (seconds) ---> 15
c-Expected LE ARP response
     time (seconds) --------------> 1
c-Flush timeout (seconds) --------> 4
c-Path switching delay (seconds) -> 6
  Local segment ID ---------------> 0
```

```
  Multicast send VCC type --------> Best Effort
  Multicast send VCC avg rate ----> 0
  Multicast send VCC peak rate ---> 0
c-Connection complete timer (sec)-> 4
  LE client ATM address ----------> 39999999999999000011110221000 4AC20E695F1
  LE client identifier -----------> 4
  LE client current state --------> Operational
  Last failure response code -----> None
  Last failure state -------------> Initial State
  Protocol -----------------------> 2
  LE protocol version ------------> 1
  Topology change ----------------> False
  Configuration server ATM address> 47007900000000000000000000A03E00000100
  Configuration source -----------> Used well known address
  Actual LAN type ----------------> TokenRing
  Actual max data frame size -----> 4544
  Actual ELAN name ---------------> TKR_1
  Actual LE server ATM address ---> 399999999999990000111101114000821000 0107
  Proxy --------------------------> False
  LE ARP requests outbound -------> 18
  LE ARP requests inbound --------> 0
  LE ARP replies outbound --------> 0
  LE ARP replies inbound ---------> 3
  Control frames outbound --------> 23
  Control frames inbound ---------> 67
  SVC failures -------------------> 0
  Configuration direct interface -> 0
  Configuration direct VPI -------> 0
  Configuration direct VCI -------> 0
  Control direct interface -------> 0
  Control direct VPI -------------> 0
  Control direct VCI -------------> 76
  Control distribute interface ---> 0
  Control distribute VPI ---------> 0
  Control distribute VCI ---------> 78
  Multicast send interface -------> 0
  Multicast send VPI -------------> 0
  Multicast send VCI -------------> 80
  Multicast forward interface ----> 0
  Multicast forward VPI ----------> 0
  Multicast forward VCI ----------> 82
p-Active MAC address -------------> 40009672E001
s-Client enabled state -----------> Enabled
c-Best effort peak rate (Mb/sec)--> 155
  Max LEC connections ------------> 1024

s-Inactivity timer (Ti) ----------> Disabled
s-Response timer (T1) ------------>  0.00 seconds
s-Acknowledgement timer (T2) ----->  0.00 seconds
  Maximum stations ---------------> 0
  Maximum SAPs -------------------> 0
s-Max I frames before ack (N3) ---> 0
s-Maximum transmit window (TW) ---> 0


********************************************************************
* Information for OSA-2 CHPID E0 emulated port 1                   *
* Settable port parameters (using SET_PARM) are preceded by 's-'   *
```

```
* Configurable port parameters (using CONFIG_OSA) are preceded by 'c-' *
* Settable and configurable port parameters are preceded by 'p-'       *
************************************************************************
  Port type ----------------------> Ethernet (ATM LAN emulation)
s-Group address 1 ----------------> not set
s-Group address 2 ----------------> not set
s-Group address 3 ----------------> not set
s-Group address 4 ----------------> not set
s-Group address 5 ----------------> not set
s-Group address 6 ----------------> not set
s-Group address 7 ----------------> not set
s-Group address 8 ----------------> not set
s-Group address 9 ----------------> not set
s-Group address 10 ---------------> not set
s-Group address 11 ---------------> not set
s-Group address 12 ---------------> not set
s-Group address 13 ---------------> not set
s-Group address 14 ---------------> not set
s-Group address 15 ---------------> not set
s-Group address 16 ---------------> not set
s-Group address 17 ---------------> not set
s-Group address 18 ---------------> not set
s-Group address 19 ---------------> not set
s-Group address 20 ---------------> not set
s-Group address 21 ---------------> not set
s-Group address 22 ---------------> not set
s-Group address 23 ---------------> not set
s-Group address 24 ---------------> not set
s-Group address 25 ---------------> not set
s-Group address 26 ---------------> not set
s-Group address 27 ---------------> not set
s-Group address 28 ---------------> not set
s-Group address 29 ---------------> not set
s-Group address 30 ---------------> not set
s-Group address 31 ---------------> not set
s-Group address 32 ---------------> not set

  Multicast information

  LP number      Unit address         Multicast          Group
  (LP name)      (device number)      IP address         address

  4 (A4      )  02(20A2),03(20A3)   224.000.000.001    01005E000001
  4 (A4      )  02(20A2),03(20A3)   224.000.000.005    01005E000005
  4 (A4      )  02(20A2),03(20A3)   224.000.000.006    01005E000006


s-User data ---------------------->
  LE PDU octets inbound ----------> 025222
  Non-error LE PDU discards
     inbound ---------------------> 80
  Errored LE PDU discards inbound-> 0
  Unknown protocol LE PDU
     discards inbound ------------> 0
  LE PDU octets outbound ---------> 0920
  Non-error LE PDU discards
     outbound --------------------> 0
  Errored LE PDU discards
```

```
     outbound --------------------> 0
  Configuration mode -------------> Automatic
c-Configured LAN type ------------> Ethernet
c-Configured max data frame size--> 1516
c-Configured ELAN name -----------> ETH_1
c-Configured LE server ATM
     address ----------------------> 00000000000000000000000000000000000000000000
c-Control timeout ----------------> 120
c-Max unknown frame count --------> 5
  Max unknown frame time (seconds)> 1
c-VCC timeout period (seconds) ---> 1200
c-Max retry count ----------------> 1
c-LE_ARP cache aging time --------> 300
c-Forward delay time (seconds) ---> 15
c-Expected LE ARP response
     time (seconds) --------------> 1
c-Flush timeout (seconds) --------> 4
c-Path switching delay (seconds) -> 6
  Local segment ID ---------------> 0
  Multicast send VCC type --------> Best Effort
  Multicast send VCC avg rate ----> 0
  Multicast send VCC peak rate ---> 0
c-Connection complete timer (sec)-> 4
  LE client ATM address ----------> 399999999999990000111102210004AC20E695F2
  LE client identifier -----------> 3
  LE client current state --------> Operational
  Last failure response code -----> None
  Last failure state -------------> Initial State
  Protocol -----------------------> 1
  LE protocol version ------------> 1
  Topology change ----------------> False
  Configuration server ATM address> 47007900000000000000000000A03E00000100
  Configuration source -----------> Used well known address
  Actual LAN type ----------------> Ethernet
  Actual max data frame size -----> 1516
  Actual ELAN name ---------------> ETH_1
  Actual LE server ATM address ---> 399999999999990000111101114000821000010 9
  Proxy --------------------------> False
  LE ARP requests outbound -------> 2
  LE ARP requests inbound --------> 0
  LE ARP replies outbound --------> 0
  LE ARP replies inbound ---------> 3
  Control frames outbound --------> 7
  Control frames inbound ---------> 39
  SVC failures -------------------> 0
  Configuration direct interface -> 0
  Configuration direct VPI -------> 0
  Configuration direct VCI -------> 0
  Control direct interface -------> 0
  Control direct VPI -------------> 0
  Control direct VCI -------------> 77
  Control distribute interface ---> 0
  Control distribute VPI ---------> 0
  Control distribute VCI ---------> 79
  Multicast send interface -------> 0
  Multicast send VPI -------------> 0
  Multicast send VCI -------------> 81
  Multicast forward interface ----> 0
```

```
  Multicast forward VPI ----------> 0
  Multicast forward VCI ----------> 83
p-Active MAC address -------------> 02009672E002
s-Client enabled state -----------> Enabled
c-Best effort peak rate (Mb/sec)--> 155
  Max LEC connections ------------> 1024

s-Inactivity timer (Ti) ----------> Disabled
s-Response timer (T1) ------------>  0.00 seconds
s-Acknowledgement timer (T2) ----->  0.00 seconds
  Maximum stations ---------------> 0
  Maximum SAPs -------------------> 0
s-Max I frames before ack (N3) ---> 0
s-Maximum transmit window (TW) ---> 0


*****************************************************
*           Device information starts here          *
*****************************************************
There is data for 16 device(s) on CHPID E0 on this LP

Device#  Unit address  Device state

  20A0         00       Online & allocated
  20A1         01       Online & allocated
  20A2         02       Online & allocated
  20A3         03       Online & allocated
  20A4         04       Offline
  20A5         05       Offline
  20A6         06       Offline
  20A7         07       Offline
  20A8         08       Offline
  20A9         09       Offline
  20AA         0A       Offline
  20AB         0B       Offline
  20AC         0C       Offline
  20AD         0D       Offline
  20AE         0E       Offline
  20AF         FE       Online

************************************************************************
***              Start of OSA address table for CHPID E0             ***
************************************************************************
* UA(Dev) Mode     Port     Entry specific information      Entry  Valid
************************************************************************
                              LP 1 (A1      )
00(20A0) N/A                                                  N/A    CSS
01(20A1) N/A                                                  N/A    CSS
02(20A2) N/A                                                  N/A    CSS
03(20A3) N/A                                                  N/A    CSS
04(20A4) N/A                                                  N/A    CSS
05(20A5) N/A                                                  N/A    CSS
06(20A6) N/A                                                  N/A    CSS
07(20A7) N/A                                                  N/A    CSS
08(20A8) N/A                                                  N/A    CSS
09(20A9) N/A                                                  N/A    CSS
0A(20AA) N/A                                                  N/A    CSS
0B(20AB) N/A                                                  N/A    CSS
```

```
0C(20AC) N/A                                                    N/A    CSS
0D(20AD) N/A                                                    N/A    CSS
0E(20AE) N/A                                                    N/A    CSS

************************************************************************
                            LP 2 (A2      )
00(20A0) N/A                                                    N/A    CSS
01(20A1) N/A                                                    N/A    CSS
02(20A2) N/A                                                    N/A    CSS
03(20A3) N/A                                                    N/A    CSS
04(20A4) N/A                                                    N/A    CSS
05(20A5) N/A                                                    N/A    CSS
06(20A6) N/A                                                    N/A    CSS
07(20A7) N/A                                                    N/A    CSS
08(20A8) N/A                                                    N/A    CSS
09(20A9) N/A                                                    N/A    CSS
0A(20AA) N/A                                                    N/A    CSS
0B(20AB) N/A                                                    N/A    CSS
0C(20AC) N/A                                                    N/A    CSS
0D(20AD) N/A                                                    N/A    CSS
0E(20AE) N/A                                                    N/A    CSS

************************************************************************
                            LP 3 (A3      )
00(20A0) N/A                                                    N/A    CSS
01(20A1) N/A                                                    N/A    CSS
02(20A2) N/A                                                    N/A    CSS
03(20A3) N/A                                                    N/A    CSS
04(20A4) N/A                                                    N/A    CSS
05(20A5) N/A                                                    N/A    CSS
06(20A6) N/A                                                    N/A    CSS
07(20A7) N/A                                                    N/A    CSS
08(20A8) N/A                                                    N/A    CSS
09(20A9) N/A                                                    N/A    CSS
0A(20AA) N/A                                                    N/A    CSS
0B(20AB) N/A                                                    N/A    CSS
0C(20AC) N/A                                                    N/A    CSS
0D(20AD) N/A                                                    N/A    CSS
0E(20AE) N/A                                                    N/A    CSS

************************************************************************
                            LP 4 (A4      )
00(20A0) passthru   00  PRI 192.168.140.012                     SIU    ALL
01(20A1) passthru   00  PRI 192.168.140.012                     SIU    ALL
02(20A2) passthru   01  PRI 192.168.140.075                     SIU    ALL
03(20A3) passthru   01  PRI 192.168.140.075                     SIU    ALL
04(20A4) N/A                                                    N/A    CSS
05(20A5) N/A                                                    N/A    CSS
06(20A6) N/A                                                    N/A    CSS
07(20A7) N/A                                                    N/A    CSS
08(20A8) N/A                                                    N/A    CSS
09(20A9) N/A                                                    N/A    CSS
0A(20AA) N/A                                                    N/A    CSS
0B(20AB) N/A                                                    N/A    CSS
0C(20AC) N/A                                                    N/A    CSS
0D(20AD) N/A                                                    N/A    CSS
0E(20AE) N/A                                                    N/A    CSS
```

```
**************************************************************************
                            LP 5 (A5      )
00(20A0) N/A                                                  N/A    CSS
01(20A1) N/A                                                  N/A    CSS
02(20A2) N/A                                                  N/A    CSS
03(20A3) N/A                                                  N/A    CSS
04(20A4) N/A                                                  N/A    CSS
05(20A5) N/A                                                  N/A    CSS
06(20A6) N/A                                                  N/A    CSS
07(20A7) N/A                                                  N/A    CSS
08(20A8) N/A                                                  N/A    CSS
09(20A9) N/A                                                  N/A    CSS
0A(20AA) N/A                                                  N/A    CSS
0B(20AB) N/A                                                  N/A    CSS
0C(20AC) N/A                                                  N/A    CSS
0D(20AD) N/A                                                  N/A    CSS
0E(20AE) N/A                                                  N/A    CSS

**************************************************************************
                            LP 8 (A8      )
00(20A0) N/A                                                  N/A    CSS
01(20A1) N/A                                                  N/A    CSS
02(20A2) N/A                                                  N/A    CSS
03(20A3) N/A                                                  N/A    CSS
04(20A4) N/A                                                  N/A    CSS
05(20A5) N/A                                                  N/A    CSS
06(20A6) N/A                                                  N/A    CSS
07(20A7) N/A                                                  N/A    CSS
08(20A8) N/A                                                  N/A    CSS
09(20A9) N/A                                                  N/A    CSS
0A(20AA) N/A                                                  N/A    CSS
0B(20AB) N/A                                                  N/A    CSS
0C(20AC) N/A                                                  N/A    CSS
0D(20AD) N/A                                                  N/A    CSS
0E(20AE) N/A                                                  N/A    CSS

**************************************************************************
                            LP 9 (A9      )
00(20A0) N/A                                                  N/A    CSS
01(20A1) N/A                                                  N/A    CSS
02(20A2) N/A                                                  N/A    CSS
03(20A3) N/A                                                  N/A    CSS
04(20A4) N/A                                                  N/A    CSS
05(20A5) N/A                                                  N/A    CSS
06(20A6) N/A                                                  N/A    CSS
07(20A7) N/A                                                  N/A    CSS
08(20A8) N/A                                                  N/A    CSS
09(20A9) N/A                                                  N/A    CSS
0A(20AA) N/A                                                  N/A    CSS
0B(20AB) N/A                                                  N/A    CSS
0C(20AC) N/A                                                  N/A    CSS
0D(20AD) N/A                                                  N/A    CSS
0E(20AE) N/A                                                  N/A    CSS

**************************************************************************
                            LP A (A10     )
00(20A0) N/A                                                  N/A    CSS
01(20A1) N/A                                                  N/A    CSS
```

```
02(20A2) N/A                                                    N/A    CSS
03(20A3) N/A                                                    N/A    CSS
04(20A4) N/A                                                    N/A    CSS
05(20A5) N/A                                                    N/A    CSS
06(20A6) N/A                                                    N/A    CSS
07(20A7) N/A                                                    N/A    CSS
08(20A8) N/A                                                    N/A    CSS
09(20A9) N/A                                                    N/A    CSS
0A(20AA) N/A                                                    N/A    CSS
0B(20AB) N/A                                                    N/A    CSS
0C(20AC) N/A                                                    N/A    CSS
0D(20AD) N/A                                                    N/A    CSS
0E(20AE) N/A                                                    N/A    CSS

********************************************************************************
***                    Legend for abbreviations                              ***
********************************************************************************
* Entry column             Valid column
* ------------             ------------
* S   - Started            OSA - Does not exist in IOCDS, but is on OSA
* NS  - Not started        CSS - Exists only in Channel Subsystem (IOCDS)
* SIU - Started & in use   ALL - Exists on the OSA and in IOCDS
* N/A - Not Applicable
* REJ - Rejected (see messages at top of OSA address table)
*
* Entry Specific Information
* --------------------------
* Passthru           - Default entry, Home IP address & Netmask
* SNA                - VTAM IDNUM if port number is FF
* MPC (IP or IPX)    - OSA name
* MPC (QDIO Control) - OSA name
* MPC (QDIO Data)    - Default entry & OSA name
********************************************************************************
********************************************************************************
***                       End of OAT entries                                 ***
********************************************************************************

********************************************************************************
**********************     End of Query data    **********************
```

E.7 RODM configuration files

E.7.1 USER.PROCLIB(EKGXRODM)

```
//RODM    PROC  TYPE=W,
//        NAME=,
//        INIT=,
//        CLRSSB=NO,
//        CUST=EKGCUST,
//        ARM=*NOARM,            ** AUTOMATIC RESTART MANAGER OPTION
//        SUBSYM=*SUBSYM         ** SYMBOL SUBSTITUTION OPTION
//*******************************************************************
//START    EXEC PGM=EKGTC000,REGION=0K,TIME=1440,
// PARM='&TYPE,&NAME,&INIT,&CLRSSB,&CUST,&ARM,&SUBSYM'
//STEPLIB  DD DSN=NETVIEW.V1R3M0.SEKGMOD1,DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.SEKGMOD2,DISP=SHR
//*        DD DSN=NETVIEW.V1R3USER.RA03N.USERLNK,DISP=SHR
```

```
//         DD DSN=CEE.SCEERUN,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSDUMP  DD SYSOUT=*
//SYSTERM  DD SYSOUT=*
//EKGLOGP  DD DSN=NETVIEW.V1R3USER.RA03N.EKGLOGP,DISP=SHR,
//         AMP=('BUFND=10')
//EKGLOGS  DD DSN=NETVIEW.V1R3USER.RA03N.EKGLOGS,DISP=SHR,
//         AMP=('BUFND=10')
//EKGCUST  DD DSN=NETVIEW.V1R3M0.SEKGSMP1,DISP=SHR
//EKGLANG  DD DSN=NETVIEW.V1R3M0.SEKGLANG,DISP=SHR
//*
//*********************************************************************
//*   RODM CHECKPOINT DATASETS                                        *
//*                                                                   *
//*   THE SIZE OF THE CHECKPOINT DATASETS MUST BE LARGE ENOUGH TO     *
//*   CHECKPOINT ALL OF THE DATA THAT IS CREATED IN RODM.  IF THERE   *
//*   IS NOT ENOUGH SPACE IN THE CHECKPOINT DATASETS, NO MORE OBJECTS *
//*   OR CLASSES CAN BE CREATED IN RODM, EVEN IF THE DATASPACE IS NOT *
//*   FULL.                                                           *
//*                                                                   *
//*   IF YOU DO NOT USE THE RODM CHECKPOINT FUNCTION, CODE            *
//*   CHECKPOINT_FUNCTION(NONE) IN THE CUSTOMIZATION FILE AND         *
//*   COMMENT OUT THE FOLLOWING 4 STATEMENTS.                         *
//*********************************************************************
//EKGMAST  DD DSN=NETVIEW.V1R3USER.RA03N.EKGMAST,DISP=SHR
//EKGTRAN  DD DSN=NETVIEW.V1R3USER.RA03N.EKGTRAN,DISP=SHR
//EKGD001  DD DSN=NETVIEW.V1R3USER.RA03N.EKGCK001,DISP=SHR
//EKGD002  DD DSN=NETVIEW.V1R3USER.RA03N.EKGCK002,DISP=SHR
//*
//*********************************************************************
//*   UNCOMMENT THESE JCL STATEMENTS IF YOU INTEND TO LOAD            *
//*   RODM DATA STRUCTURE USING AN INITIALIZATION METHOD.             *
//*                                                                   *
//*   NOTE:  YOU CANNOT LOAD THE GMFHS/SNATM DATA MODEL               *
//*          USING AN INITIALIZATION METHOD.                          *
//*********************************************************************
//*EKGLUTB  DD DSN=NETVIEW.V1R3M0.SEKGLUTB,DISP=SHR
//*EKGPRINT DD SYSOUT=*
//*EKGIN1   DD DSN=NETVIEW.V1R3M0.SEKGSMP1(EKGIN1),DISP=SHR
//*EKGIN2   DD DSN=NETVIEW.V1R3M0.SEKGCAS1,DISP=SHR
//*EKGIN3   DD DUMMY
```

E.7.2 NETVIEW.V1R3M0.SEKGSMP1(EKGCUST)

```
/*********************************************************************/
/*  LICENSED MATERIALS - PROPERTY OF IBM                             */
/*  5697-B82 (C) COPYRIGHT TIVOLI SYSTEMS 1997, 1998                 */
/*  5655-007 (C) COPYRIGHT IBM CORP. 1992, 1995                      */
/*  ALL RIGHTS RESERVED.                                             */
/*                                                                   */
/*  US GOVERNMENT USERS RESTRICTED RIGHTS                            */
/*  - USE, DUPLICATION OR  DISCLOSURE RESTRICTED BY                  */
/*    GSA ADP SCHEDULE CONTRACT WITH IBM CORPORATION.                */
/*                                                                   */
/*  NAME(EKGCUST)  SAMPLE(EKGCUST)  RELATED-TO( )                    */
/*                                                                   */
/*  DESCRIPTION: SAMPLE RODM CUSTOMIZATION FILE                      */
```

```
/*                                                                   */
/*        NOTES: ALL THE VALUES SPECIFIED FOR EACH OPTIONAL          */
/*               CUSTOMIZATION PARAMETER ARE THE DEFAULT VALUES      */
/*               USED BY RODM.                                       */
/*                                                                   */
/*  EKGCUST CHANGE ACTIVITY:                                         */
/*                                                                   */
/*******************************************************************/

/*******************************************************************/
/* Number of asynchronous tasks.  The asynchronous tasks run       */
/* some of the methods that are triggered within RODM.             */
/*******************************************************************/
  ASYNC_TASKS           ( 5 )
/*******************************************************************/
/* Cell pool and cell size definition.  You can specify up to 200  */
/* cell sizes and cell pools.                                      */
/*******************************************************************/
 CELL_POOLS (  8, 12, 16, 20 )
 CELL_POOLS ( 24, 28, 32, 36:2 )
 CELL_POOLS ( 40:2, 48:2, 52:2, 56:2 )
 CELL_POOLS ( 60:2, 64:2, 68:3, 72:3 )
 CELL_POOLS ( 80:3, 88:3, 100:3, 104:3 )
 CELL_POOLS ( 112:3, 120:3, 128:3, 136:4 )
 CELL_POOLS ( 144:4, 152:4, 160:4, 168:4 )
 CELL_POOLS ( 176:4, 184:4, 192:4, 200:4 )
 CELL_POOLS ( 208:4, 216:4, 224:4, 232:4 )
 CELL_POOLS ( 240:4, 248:4, 256:4, 384:6 )
 CELL_POOLS ( 484:9, 512:8, 768:9, 1024:8 )
 CELL_POOLS ( 1536:12, 2048:16, 3072:24, 4096:32 )
 CELL_POOLS ( 6144:48, 8192:64, 12288:96, 16384:128 )
 CELL_POOLS ( 24576:192, 32768:256 )
/*******************************************************************/
/* Number of users that can enter RODM through the user API        */
/* concurrently.                                                   */
/*******************************************************************/
  CONCURRENT_USERS       ( 10 )
/*******************************************************************/
/* The RODM dump limit.  The first value specifies the number of   */
/* dumps that can be taken within the time interval.  The second   */
/* value specifies the number of minutes that RODM will use as a   */
/* time interval.  The default, (10,1440), means 10 dumps will     */
/* be allowed within any 24 hour period.  If more dumps occur      */
/* after the dump limit is exceeded, an informational message      */
/* is issued,(EKG2305I), and the dump is not taken.  This option   */
/* can help you limit the system resources required by RODM if     */
/* some sort of repetitive error is encountered.                   */
/*******************************************************************/
  DUMP_LIMIT             (10,1440)
/*******************************************************************/
/* Whether RODM will take a dump when it encounters problem copying */
/* user data:                                                      */
/*                                                                 */
/*      YES - take a dump if bad user data is encountered          */
/*      NO  - do not take a dump if bad user data is encountered   */
/*******************************************************************/
  DUMP_FOR_BAD_USER_DATA ( NO )
/*******************************************************************/
```

```
/* Specifies which user address spaces to be dumped when an abend   */
/* occurs in RODM:                                                  */
/*                                                                  */
/* REQUESTOR                                                        */
/*   -- dump only the user address space that is involved in the    */
/*      error                                                       */
/*                                                                  */
/* CONNECTED                                                        */
/*   -- dump all address spaces that have connected to RODM through */
/*      the user API                                                */
/********************************************************************/
  DUMP_SCOPE             (REQUESTOR)
/********************************************************************/
/* Extended PL/I heap storage size                                  */
/********************************************************************/
  EXTEND_HEAP_SIZE       ( 64K )
/********************************************************************/
/* Number of I/O queue blocks before flushing                       */
/********************************************************************/
  IO_QUEUE_THRESHOLD     ( 5  )
/********************************************************************/
/* API log level, an API call with the return code greater than     */
/* or equal to this value will be logged in the RODM log file       */
/********************************************************************/
  LOG_LEVEL              ( 8 )
/********************************************************************/
/* Maximum number of free chunks kept                               */
/********************************************************************/
  MAX_CHUNK              ( 256 )
/********************************************************************/
/* Maximum number of 16 MB segments that can be allocated in the    */
/* RODM address space.                                              */
/*                                                                  */
/*    Minimum value 3                                               */
/*    Maximum value 64                                              */
/*                                                                  */
/* IBM recommends setting this keyword to the maximum value, unless */
/* you wish to limit the capacity of RODM.  If the MAX_SEGMENT_NUM  */
/* value is reached, no more objects, links, or classes can be      */
/* created.                                                         */
/*                                                                  */
/* The 16 MB segments are allocated one-by-one as needed.           */
/*                                                                  */
/* See also the EKGDWIND and EKGWINDW samples for information about */
/* correlating this value to the size of your checkpoint datasets.  */
/********************************************************************/
  MAX_SEGMENT_NUM        ( 64  )
/********************************************************************/
/* Maximum number of 16 MB data windows in the RODM dataspace(s).   */
/*                                                                  */
/* Minimum value 4                                                  */
/* Maximum value 191                                                */
/*                                                                  */
/* IBM recommends setting this keyword to the maximum value, unless */
/* you wish to limit the capacity of RODM.  If the MAX_WINDOW_NUM   */
/* value is reached, no more objects, links or classes can be       */
/* created.                                                         */
/*                                                                  */
```

```
/* The 16 MB data windows are allocated one-by-one as needed.       */
/*                                                                  */
/* See also the EKGDWIND and EKGWINDW samples for information about */
/* correlating this value to the size of your checkpoint datasets.  */
/********************************************************************/
  MAX_WINDOW_NUM        ( 191 )
/********************************************************************/
 /*                                                                 */
 /*  CHARACTER_VALIDATION specifies what degree of checking (for    */
 /*  valid characters) should be performed for class, field and     */
 /*  object names.                                                  */
 /*                                                                 */
 /*  YES is the default, and is how character validation was        */
 /*  handled prior to the introduction of the CHARACTER_VALIDATION  */
 /*  keyword.                                                       */
 /*                                                                 */
 /*    When CHARACTER_VALIDATION(YES) is specified (or defaulted):  */
 /*                                                                 */
 /*      For class and field names:                                 */
 /*                                                                 */
 /*        1st character must be alpha-numeric                      */
 /*                                                                 */
 /*        Subsequent characters can be alpha-numeric and any of    */
 /*        the following special characters:  _ @ # .               */
 /*                                                                 */
 /*                                                                 */
 /*      For object names:                                          */
 /*                                                                 */
 /*        1st character must be alpha-numeric                      */
 /*                                                                 */
 /*        Subsequent characters can be alpha-numeric and any of    */
 /*        the following special characters:  _ @ # . , : ; + ?     */
 /*                                           & % * = < > / ' "     */
 /*                                           ( ) -                 */
 /*                                                                 */
 /*                                           (these are present    */
 /*                                            on every code page)  */
 /*                                                                 */
 /*                                                                 */
 /*                                                                 */
 /*    When CHARACTER_VALIDATION(NO) is specified:                  */
 /*                                                                 */
 /*       For class, field, AND object names:                       */
 /*                                                                 */
 /*           No blanks are allowed  (40'X)                         */
 /*           No nulls are allowed   (00'X)                         */
 /*           No pound sign as first character  ('7B'X)             */
 /*                                                                 */
 /*******************************************************************/
  CHARACTER_VALIDATION (YES)
/********************************************************************/
/* MAPI log level, a MAPI call with the return code greater than    */
/* or equal to this value will be logged in the RODM log file       */
/********************************************************************/
  MLOG_LEVEL            ( 8 )
/********************************************************************/
/* MAPI trace type                                                  */
/********************************************************************/
```

```
  MTRACE_TYPE            (X'0000007C')
/*********************************************************************/
/* PL/I ISA size                                                     */
/*********************************************************************/
  PLI_ISA                ( 40K )
/*********************************************************************/
/* Primary PL/I heap storage size                                    */
/*********************************************************************/
  PRIMARY_HEAP_SIZE      ( 64K )
/*********************************************************************/
/* Waiting time for quiescence before killing a task                 */
/* (in 0.01 second units)                                            */
/*********************************************************************/
  QUIES_WAIT_TASK_TIME   ( 200 )
/*********************************************************************/
/* Waiting time for quiescence before killing a user                 */
/* (in 0.01 second units)                                            */
/*********************************************************************/
  QUIES_WAIT_USER_TIME   ( 200 )
/*********************************************************************/
/* Number of same name system status block can concurrently exist    */
/* in the system                                                     */
/*********************************************************************/
  SSB_CHAIN              ( 3     )
/*********************************************************************/
/* Security class name                                               */
/*                                                                   */
/*     RODMMGR is the name of the RACF class defined for RODM        */
/*     in the various SAF products.  The RODMMGR class is            */
/*     available in the base of RACF 2.2, and is available on        */
/*                                                                   */
/*              RACF 1.9   with PTF UW00497                          */
/*              RACF 1.9.2 with PTF UW00498                          */
/*              RACF 2.1   with PTF UW90113                          */
/*                                                                   */
/*                                                                   */
/*     If you choose to define your own RACF class, set the          */
/*     corresponding name below.                                     */
/*                                                                   */
/*     If you do not have a SAF product installed, or you wish       */
/*     to bypass the security checking for the RODM access levels,   */
/*     use the value (*TSTRODM) for the SEC_CLASS name.              */
/*                                                                   */
/*********************************************************************/
  SEC_CLASS              ( *TSTRODM )
/*********************************************************************/
/* The prefix for the 6 resource names that will be authority        */
/* checked for various levels of access within RODM.  For example,   */
/* if "RODM" is the prefix, the resource names will be RODM1,        */
/* RODM2, RODM3, RODM4, RODM5 and RODM6.                             */
/*                                                                   */
/* The default value for SEC_RNAME is the RODM name if you do not    */
/* specify a value for this keyword below.  (The RODM name is        */
/* specified using the NAME= parameter in the RODM startup JCL,      */
/* or with the "dot" suffix.  For example, the RODM name would       */
/* be "ZZRODM" in the following commands:                            */
/*                                                                   */
/*            S EKGXRODM,NAME=ZZRODM                                 */
```

```
/*            S EKGXRODM.ZZRODM                                    */
/*                                                                 */
/* Uncomment the following statement and set the value if you      */
/* want to change the prefix of your SAF resource name to          */
/* something other than the RODM name.                             */
/*******************************************************************/
/*SEC_RNAME              ( RODM )         Security resource name    */
/*******************************************************************/
/* Whether a message is written to the system console when a       */
/* a connection request fails.  (in addition to the message        */
/* that is already written to the RODM log)                        */
/*                                                                 */
/*   YES - write message to the system console                     */
/*   NO  - Don't write message to the system console               */
/*******************************************************************/
  CONNECT_VIOLATION_MESSAGE (NO)
/*******************************************************************/
/* Number of backup segments                                       */
/*******************************************************************/
  SEGMENT_POCKETS        ( 1 )
/*******************************************************************/
/* Whether user and method API requests should be rejected or wait */
/* when a checkpoint is in progress.                               */
/*                                                                 */
/* If transparent checkpoint is enabled, user and method requests  */
/* will wait until the checkpoint completes, and run afterwards.   */
/* In this case, applications do not have to implement retry logic */
/* for checkpoint-in-progress conditions.                          */
/*                                                                 */
/* If transparent checkpoint is not enabled, user and method       */
/* API requests will be rejected with a checkpoint-in-progress     */
/* condition, and must retry.                                      */
/*                                                                 */
/*    YES -- queue up requests                                     */
/*    NO  -- do not queue up requests                              */
/*******************************************************************/
  TRANSPARENT_CHECKPOINT (NO)
/*******************************************************************/
/* Waiting time for allocating windows or segments                 */
/* (in .01 second units)                                           */
/*******************************************************************/
  WAIT_ALLOC_TIME        ( 100 )
/*******************************************************************/
/* Unit of waiting time for checkpointing a window or a segment    */
/* (in 0.01 second units)                                          */
/*******************************************************************/
  WINDOW_CHKPT_TIME      ( 1500 )
/*******************************************************************/
/* Number of backup windows                                        */
/*******************************************************************/
  WINDOW_POCKETS         ( 1 )
/*******************************************************************/
/* Indicates whether or not the checkpoint function is to be       */
/* enabled, as well as what RODM will do when the checkpoint       */
/* datasets do not exist or are too small to map the master window, */
/* translation segment(s), and/or data window(s).                  */
/*                                                                 */
/*   REQUIRE -- checkpoint function must be enabled.  If the       */
```

```
/*             master window, first translation segment, or the    */
/*             first data window cannot be mapped, then RODM will  */
/*             terminate; otherwise, RODM will limit storage usage */
/*             to that which may be saved in the checkpoint        */
/*             datasets.                                           */
/*                                                                 */
/*   REQUEST -- checkpoint function enabling is to be attempted,   */
/*             but RODM may continue if the failure to enable is   */
/*             for a reason outlined above.  This is the default.  */
/*                                                                 */
/*   NONE    -- checkpoint function is to be disabled.  This value */
/*             is only valid for a RODM cold start.                */
/*******************************************************************/
  CHECKPOINT_FUNCTION    ( REQUEST )
```

E.8 GMFHS configuration files

E.8.1 USER.PROCLIB(CNMGMFHS)

```
//GMFHS   PROC
//*
//STEP1    EXEC PGM=DUIFT000,REGION=32M,
// PARM='N,RESWS=NO,DOMAIN=RA03N,ARM=*NOARM,SUBSYM=*SUBSYM'
//STEPLIB  DD DSN=NETVIEW.V1R3M0.CNMLINK,DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.SEKGMOD1,DISP=SHR
//* LE/370
//*       DD DSN=CEE.SCEERUN,DISP=SHR
//*   INITIALIZATION PARAMETERS DATASET
//CNMPARM  DD DSN=NETVIEW.V1R3USER.RA03N.DSIPARM,DISP=SHR
//         DD DSN=NETVIEW.V1R3M0.DSIPARM,DISP=SHR
//*   GMFHS MESSAGES DATASET
//CNMMSG1  DD DSN=NETVIEW.V1R3M0.SDUIMSG1,DISP=SHR
//CNMM     DD SYSOUT=A
//CNMD     DD SYSOUT=A
//CNMI     DD SYSOUT=A
//CNMO     DD SYSOUT=A
//CNMF     DD SYSOUT=A
//CNME     DD SYSOUT=A
//CNMV     DD SYSOUT=A
//CNMC     DD SYSOUT=A
//CNMS     DD SYSOUT=A
//CNMT     DD SYSOUT=A
//CNMP     DD SYSOUT=A
//CNMR     DD SYSOUT=A
//*   RUNTIME LIBRARY MESSAGES
//SYSTERM  DD SYSOUT=A
//*   THE FOLLOWING ARE FOR THE RODM LOAD UTILITY
//EKGLANG  DD DSN=NETVIEW.V1R3M0.SEKGLANG,DISP=SHR
//EKGLUTB  DD DSN=NETVIEW.V1R3M0.SEKGLUTB,DISP=SHR
//EKGPRINT DD SYSOUT=A
//EKGIN3   DD DUMMY,DCB=BLKSIZE=80
//*
//* NOTE: REMOVE THE COMMENTS FROM THE NEXT TWO STATEMENTS ALONG WITH THE
//*       THE JOBCARD WHEN MAKING THIS A BATCH JOB.
//*        PEND
//*GMFHS   EXEC GMFHS
```

E.8.2 NETVIEW.V1R3USER.RA03N.DSIPARM(DUIGINIT)

```
*********************************************************************
*  Licensed Materials - Property of Tivoli Systems
*  5697-B82 (C) COPYRIGHT Tivoli SYSTEMS  1997, 1999
*  5655-007 (C) COPYRIGHT IBM CORP. 1993, 1995
*  ALL RIGHTS RESERVED.
*
*  US Government Users Restricted Rights - Use, duplication or
*  disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
*********************************************************************
*  NAME(DUIGINIT)  SAMPLE(DUIGINIT)  REALTED-TO( )
*  DESCRIPTION: PROVIDES INITIALIZATION PARAMETERS TO THE GMFHS
*               HOST MAIN TASK
*********************************************************************
*   '*' in column 1 is a comment
*********************************************************************
*GMTOFFSET=+tttt
TRACE=OFF
*TASK=(DBSERVER,VIEWMGR)
*LEVEL=10
*API=(RODM,IPC,PPI)
*TYPE=(PDU,SCO,PPI,GDS,CNMTAMEL,NOTIFY)
*STORAGE=YES
*PRINTPDU38=INTERNAL
*TRACEPAGES=100
*TRACEBYTES=0
*********************************************************************
*     RODM Name
RODMNAME=RODMNAME
*********************************************************************
*     RODM access RACF userid
RODMID=LUCY
*********************************************************************
*     Domain Identifier
DOMAIN=RA03N
*********************************************************************
*CHECKPOINT=(STARTUP,CONFIG,TERM)
*********************************************************************
*LCON-NMG-POLL-INTERVAL=18000
*LCON-NCC-RSC-LIMIT=10
*LCON-STATUS-DELAY-TIME=50
*LCON-STATUS-DELAY-MAX=10
*LCON-REPORT-UNKNOWN-STATUS=00
*LCON-HEX-SUBVECTOR-DISPLAY=00
*LCON-OPERATOR-CMD-AUDIT=00
*LCON-MAX-QUEUE-IPC=10000
*LCON-MAX-QUEUE-OPERIF=10000
*LCON-MAX-QUEUE-DBSERVER=10000
*LCON-MAX-QUEUE-NETCON=10000
*LCON-MAX-QUEUE-EVENTMGR=10000
*LCON-MAX-QUEUE-VIEWMGR=10000
*LCON-MAX-QUEUE-VSTATMGR=10000
*LCON-MAX-QUEUE-NETCMD=10000
*LCON-MAX-QUEUE-MAINTASK=10000
*LCON-MAX-QUEUE-RTMGR=10000
*LCON-MAX-QUEUE-IRMGR=10000
*LCON-AGGRST-REQUIRED=NO
```

```
*LCON-EVCHANGE-BUFFER-INTERVAL=500
*LCON-AIP-RESET-INTERVAL=12000
*LCON-AGG-BUNDLE-INTERVAL=500
*LCON-SNATM-TIMEOUT=12000
*LCON-ALERT-CMD-TIMEOUT=30000
*LCON-MAX-LOCATE-RESOURCE-VIEWS=12
*LCON-ASSOCIATE-NULL-NODE-WITH-LINK=0
```

E.9 NetView procedure configuration files

E.9.1 USER.PROCLIB(NETVA03)

```
//CNMAPROC PROC
//NETVIEW  EXEC PGM=BNJLINTX,TIME=1440,
//         REGION=50M,
//         PARM=(24K,200,'RA03N','','*NOARM',''),
//         DPRTY=(13,13)
//STEPLIB  DD   DSN=NETVIEW.V1R3M0.CNMLINK,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEKGMOD1,DISP=SHR
//         DD   DSN=REXX.V1R3M0.SEAGLMD,DISP=SHR
//         DD   DSN=REXX.V1R3M0.SEAGALT,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEZLLINK,DISP=SHR
//SYSTCPD  DD DSN=TCPIP.TCPPARMS(TCPD&SYSCLONE.A),DISP=SHR
//DSICLD   DD   DSN=NETVIEW.V1R3USER.RA03N.CNMCLST,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.CNMCLST,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEZLCLST,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.CNMSAMP,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEKGSMP1,DISP=SHR
//DSIOPEN  DD   DSN=NETVIEW.V1R3USER.RA03N.SDSIOPEN,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SDSIOPEN,DISP=SHR
//DSIPARM  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIPARM,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.DSIPARM,DISP=SHR
//DSILIST  DD   DSN=NETVIEW.V1R3USER.RA03N.DSILIST,DISP=SHR
//DSIASRC  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIASRC,DISP=SHR
//DSIARPT  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIARPT,DISP=SHR
//DSIVTAM  DD   DSN=NETVIEW.V1R3USER.RA03N.VTAMLST,DISP=SHR
//         DD   DSN=SYS1.VTAMLST,DISP=SHR
//DSIPRF   DD   DSN=NETVIEW.V1R3USER.RA03N.DSIPRF,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.DSIPRF,DISP=SHR
//DSIMSG   DD   DSN=NETVIEW.V1R3M0.SDSIMSG1,DISP=SHR
//BNJPNL1  DD   DSN=NETVIEW.V1R3M0.BNJPNL1,DISP=SHR
//*BNJPNL1  DD   DSN=NETVIEW.V1R3M0.SBNJPNL3,DISP=SHR
//BNJPNL2  DD   DSN=NETVIEW.V1R3M0.BNJPNL2,DISP=SHR
//CNMPNL1  DD   DSN=NETVIEW.V1R3M0.CNMPNL1,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEKGPNL1,DISP=SHR
//         DD   DSN=NETVIEW.V1R3USER.RA03N.SEZLPNLU,DISP=SHR
//         DD   DSN=NETVIEW.V1R3M0.SEZLPNLU,DISP=SHR
//DSILOGP  DD   DSN=NETVIEW.V1R3USER.RA03N.DSILOGP,
//         DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSILOGS  DD   DSN=NETVIEW.V1R3USER.RA03N.DSILOGS,
//         DISP=SHR,AMP='AMORG,BUFNI=20,BUFND=20'
//DSITRCP  DD   DSN=NETVIEW.V1R3USER.RA03N.DSITRCP,
//         DISP=SHR,AMP=AMORG
//DSITRCS  DD   DSN=NETVIEW.V1R3USER.RA03N.DSITRCS,
//         DISP=SHR,AMP=AMORG
//AAUVSPL  DD   DSN=NETVIEW.V1R3USER.RA03N.AAUVSPL,
//         DISP=SHR,AMP='AMORG'
```

```
//AAUVSSL  DD   DSN=NETVIEW.V1R3USER.RA03N.AAUVSSL,
//         DISP=SHR,AMP='AMORG'
//BNJLGPR  DD   DSN=NETVIEW.V1R3USER.RA03N.BNJLGPR,
//         DISP=SHR,AMP='AMORG'
//BNJLGSE  DD   DSN=NETVIEW.V1R3USER.RA03N.BNJLGSE,
//         DISP=SHR,AMP='AMORG'
//DSIKPNL  DD   DSN=NETVIEW.V1R3USER.RA03N.DSIKPNL,
//         DISP=SHR,AMP=AMORG
//DSISVRT  DD   DSN=NETVIEW.V1R3USER.RA03N.DSISVRT,
//         DISP=SHR,AMP=AMORG
//EZLSTAT  DD   DSN=NETVIEW.V13USER.RA03N.STATS,
//         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//EZLLOGP  DD   DSN=NETVIEW.V13USER.RA03N.LOGP,
//         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//EZLLOGS  DD   DSN=NETVIEW.V13USER.RA03N.LOGS,
//         DISP=SHR,AMP='AMORG,BUFNI=10,BUFND=5'
//SYSPRINT DD   SYSOUT=*
```

E.9.2 NETVIEW.V1R3USER.RA03N.DSIPARM(FKXCFG01)

```
*********************************************************************
*  WARNING!! All comment lines must have a * in column 1, including   *
*            blank lines                                             *
*********************************************************************
* Trap Auto Oper                                                @D8A  *
AUTOOPS    TRAPOPER,ID=AUTTRAP
*********************************************************************
INSTALLOPT TCPIP,
             INITIALIZE=Y,
             IC=FKXEINIT,
             DEFTAB=FKXTABLE,
             OPER=TCPOPER
*
INSTALLOPT IP390,
             INITIALIZE=Y
*
INSTALLOPT AON,
             INITIALIZE=Y
*DDFGENERIC TCPIP.RESTYPE,LEVELUP=SYSTEM,OTHER=COMPONEN,
*     VALUE=(NVAIX,IP390,SP,HOST,ROUTER,LINK,INFC,NAMESERV,PORT,TN3270)
*-------------------------------------------------------------------*
* AONTCP AUTOTASKS                                                  *
*-------------------------------------------------------------------*
AUTOOPS    TCPOPER,ID=(AUTTCP,10)
*-------------------------------------------------------------------*
* SET LARGE SCALE THRESHOLDING VALUES FOR NAMESERVER FAILURES (total)  *
*-------------------------------------------------------------------*
*LSTHRESH NAMESERV,COUNT=(10,00:10),EVENTLIM=50,NTFY=THR
*********************************************************************
*======== NOTIFICATION POLICY FOR NAMESERVER THRESHOLD :        ========
*========  ISSUE ALERTS AND UPDATE DDF.                         ========
*NOTIFY NAMESERV,ALERT=YES,MSG=NO,DDF=YES,INFORM=NO
*========                                                       ========
*
*-------------------------------------------------------------------*
* Define Recovery Monitoring Interval(s) & Notification Action         *
*-------------------------------------------------------------------*
*
```

```
*MONIT HOST,INTVL=(00:02,Y),INTVL=(00:15,Y),INTVL=(01:10,Y),
*           INTVL=(01:00,YF)
MONIT IPHOST,INTVL=(00:02,Y),INTVL=(00:15,Y),INTVL=(01:10,Y),
          INTVL=(01:00,YF)
*MONIT INFC,INTVL=(00:02,Y),INTVL=(00:15,N),INTVL=(01:13,YF)
*MONIT IPINFC,INTVL=(00:02,Y),INTVL=(00:15,N),INTVL=(01:13,YF)
*MONIT NAMESERV,INTVL=(00:02,Y),INTVL=(01:05,YF)
*MONIT IPNAMESERV,INTVL=(00:02,Y),INTVL=(01:05,YF)
MONIT SP,INTVL=(00:30,YF)
MONIT IPPORT,INTVL=(00:02,Y),INTVL=(00:05,Y),INTVL=(00:10,YF)
*
*--------------------------------------------------------------------*
* SET ACTIVE MONITORING FOR TCP390 OPTION                            *
*--------------------------------------------------------------------*
*
ACTMON IP390,OPTION=IP390,INTVL=01:00,STATUS=NORMAL
ACTMON STACKS,OPTION=IP390,INTVL=00:30,STATUS=NORMAL,
       FORMAT=SNMP
ACTMON ALLHOSTS,OPTION=IP390,INTVL=00:30,STATUS=NORMAL,
       FORMAT=SNMP
*ACTMON IPNAMESERV,OPTION=IP390,INTVL=00:15,STATUS=NORMAL,
*       FORMAT=PING
ACTMON SOCKETS,OPTION=IP390,INTVL=00:30,STATUS=NORMAL,
       FORMAT=PORT
*ACTMON ROUTERS,OPTION=IP390,INTVL=00:10,STATUS=NORMAL,
*       FORMAT=SNMP
*
*--------------------------------------------------------------------*
* SET DEFAULTS FOR TCP390 PING & SNMP                                *
*--------------------------------------------------------------------*
*
TCP390      DEFAULTS,PINGCNT=3,
                     PINGRETRY=3,
                     PINGLEN=64,
                     PINGTIME=10,
                     DEBUG=0,
                     VERBOSE=Y,
                     SNMPRETRY=2,
                     SNMPTO=3,
                     MAXREP=10,
                     NONREP=0,
                     RPLENGTH=64,
                     RPTO=5
*
*--------------------------------------------------------------------*
* SET SPECIFIC TCP390 ENTRIES FOR MVS STACKS                         *
*--------------------------------------------------------------------*
*  TCP 3.4 LOCAL UNIX SERVICE POINT
TCP390      NMPNET03,
                     IPADDR=9.24.104.113,
                     HIER2=SP-APPL,
                     HIER3=NETSP,
                     DOMAIN=LOCAL,
                     TCPNAME=TCPIPA,
                     UNIXSERV=YES,
                     FORMAT=STACK,
                     SNMP=MVS,
                     HOSTNAME=mvs03a
```

```
*                   SERVER=LUCY,
*
*  TCP 3.4 LOCAL TSO SERVICE POINT
*
**TCP390      NMPNET10A,
**                    IPADDR=123.123.123.123,
**                    HIER2=SP-APPL,
**                    HIER3=NETSP,
**                    DOMAIN=LOCAL,
**                    SERVER=(NV2TS,5),
**                    TCPNAME=TCP34A,
**                    FORMAT=STACK,
**                    SNMP=MVS,
**                    HOSTNAME=NMPNET10A.raleigh.test.com
*
*  TCP 3.4 REMOTE TSO SERVICE POINT
*
**TCP390      NMPNET25,
**                    IPADDR=123.123.123.123,
**                    HIER2=SP-APPL,
**                    HIER3=NETSP,
**                    DOMAIN=CNM99,
**                    TCPNAME=TCP32,
**                    FORMAT=STACK,
**                    SNMP=MVS,
**                    HOSTNAME=NMPNET25.raleigh.test.com
*
*--------------------------------------------------------------------*
* Set up TSO Servers for Local Stacks                                *
*--------------------------------------------------------------------*
*
**TSOSERV NV2TS,PROC=CNMSJTSO
**TSOSERV NV2TB,PROC=CNMSJTSB
*
*--------------------------------------------------------------------*
* SET ACTIVE MONITORING FOR TCP/IP RESOURCES USING MVS               *
*  New for R3 - do not use ACTMON & TCPIP statements for MVS         *
*               resources.  See the Admin Ref manual for further info. *
*--------------------------------------------------------------------*
*
* Dummy Nameserver
*IPNAMESERV DUMMYSRV,OPTION=IP390,SP=NMPNET03,
*           FORMAT=PING,
*           INTVL=01:00,
*           IPADDR=9.37.99.255,
*           HOSTNAME=DUMMYSRV
* Gulliver - uses ALLHOSTS ACTMON statement for STATUS, INTVL, etc.
IPHOST MVS03A,OPTION=IP390,SP=NMPNET03,
       ACTMON=STACKS,
       MIBVAR1=(tcpActiveOpens.0,LT,5000),
       MIBVAR2=(tcpInErrs.0,GT,5),
       MIBVAR3=(ipInDiscards.0,EQ,1),
       MIBVAR4=(ipRoutingDiscards.0,GE,3),
       STATUS=(NORMAL,TRESH*,DEGR*),
       HOSTNAME=MVS03A
*      IPADDR=9.24.104.113
*
IPHOST KENAIX,OPTION=IP390,SP=NMPNET03,
```

```
       ACTMON=ALLHOSTS,
       IPADDR=9.24.104.26
*      HOSTNAME=mvs03a
*
*--------------------------------------------------------------------*
* SAMPLE SOCKET DEFINITIONS                                  10@D5A  *
*--------------------------------------------------------------------*
*
IPPORT DSIWBTSK,SP=NMPNET03,
                PORT=9998,
                PROTOCOL=TCP,
                TCPNAME=TCPIPA,
                CMDTYPE=NETV,
                INTVL=00:10,
                STATUS=NORMAL,
                FORMAT=PORT,
                ACTIVATE="START TASK=DSIWBTSK",
                DESC="IBM HTTP Server"
IPPORT DSITCPIP,SP=NMPNET03,
                PORT=9999,
                PROTOCOL=TCP,
                TCPNAME=TCPIPA,
                CMDTYPE=NETV,
                INTVL=00:10,
                STATUS=NORMAL,
                FORMAT=PORT,
                SESSTAT=NO,
                ACTIVATE="START TASK=DSITCPIP",
                DESC="NetView TCP/IP Socket"
*IPPORT TELNET,SP=NMPNET10,
*              PORT=623,
*              PROTOCOL=TCP,
*              TCPNAME=INETD1,
*              INTVL=00:10,
*              FORMAT=PORT,
*              DESC="OE Telnet Socket"
*
* For CS/390 V2R6 or later Multiple TN3270 Socket support:
*     change PORT=23 to PORT=*
* This will cause AON to dynamically determine your TN3270 sockets
*
IPPORT TN3270,SP=NMPNET03,
                PORT=23,
                PROTOCOL=TCP,
                TCPNAME=TCPIPA,
                INTVL=00:10,
                FORMAT=PORT,
                STATUS=NORMAL,
                DESC="Telnet Socket Pointer"
*
IPPORT FTPSRV2,SP=NMPNET03,
             PORT=21,
             PROTOCOL=TCP,
             TCPNAME=FTPDA1,
             INTVL=00:10,
             FORMAT=PORT,
             STATUS=NORMAL,
             DESC="FTP Server on RA03"
```

```
*
*IPPORT SMTP,SP=NMPNET10,
*            PORT=25,
*            PROTOCOL=TCP,
*            TCPNAME=SMTP32,
*            INTVL=00:05,
*            FORMAT=PORT,
*            DESC="SMTP Socket 25"
*
*IPPORT CICSTCP,SP=NMPNET10,
*               PORT=3000,
*               PROTOCOL=TCP,
*               TCPNAME=TCP34,
*               INTVL=00:15,
*               FORMAT=PORT,
*               DESC="CICS TCP Server"
*
*IPPORT SNMPTRAP,SP=NMPNET03,
*                PORT=162,
*                PROTOCOL=TCP,
*                TCPNAME=OSNMPTRP,
*                INTVL=00:15,
*                FORMAT=PORT,
*                SESSTAT=NO,
*                DESC="osnmp UNIX command to receive traps"
*
*--------------------------------------------------------------------*
* DDF STATUS'S
*--------------------------------------------------------------------*
DDF CPUTHR*,PR=360,CLEAR=Y
DDF DISKTHR*,PR=360,CLEAR=Y
DDF AUTHTHR*,PR=360,CLEAR=Y
DDF PINGTHR*,PR=360,CLEAR=Y
DDF NSTHR*,PR=360,CLEAR=Y
DDF LINKDOWN*,PR=210,CLEAR=Y
DDF DEGRADED,PR=230,CLEAR=Y
DDF THRESH*,PR=330,CLEAR=Y
* Following line added by @01A
DDF NORMAL,PR=550,CLEAR=Y,REQ=NOADD
*--------------------------------------------------------------------*
* RESIDENT PROGRAMS
*--------------------------------------------------------------------*
RESIDENT FKXEACT2
RESIDENT FKXEAID1
RESIDENT FKXEAID2
RESIDENT FKXECATV
RESIDENT FKXEHNDE
RESIDENT FKXEPING
RESIDENT FKXESVPT
RESIDENT FKXEOTHR
RESIDENT FKXEALRT
RESIDENT FKXEAMS1
RESIDENT FKXENSTH
RESIDENT FKXERDWN
RESIDENT FKXERSPP
RESIDENT FKXEAIDA
RESIDENT FKXETCMD
RESIDENT FKXEAIDB
```

```
RESIDENT FKXEUCMD
RESIDENT FKXEICMD
* Session Stat Data Collector                                  @P4A  *
RESIDENT FKXETCOL
* SNMP COMMAND PROCESSOR                                       @D6A  *
RESIDENT FKXESCMD
* ADDITIONAL MONITORING ROUTINES                              6@D7A  *
RESIDENT FKXEACT1
RESIDENT FKXEACTP
RESIDENT FKXEACTT
RESIDENT FKXEACTN
RESIDENT FKXEACTS
RESIDENT FKXE390A
* SOCKET AUTOMATION ROUTINES                                  3@D5A  *
RESIDENT FKXEAIDC
RESIDENT FKXEAIDD
RESIDENT FKXEPORT
* TRAP AUTOMATION ROUTINES                                     @D8A  *
RESIDENT FKXETRAP
*
```

E.9.3 /etc/netview/ipdiscovery.conf

```
#
#Mon 1 Feb 1999 18:15-odd EST
#
# dumpFile=/quanta/data/ipd.dump
logFile=/tmp/discovery.log
snmpFile=/etc/netview/snmp.conf
scope=*.*.*.*
seeds=9.24.104.113,9.24.104.149
debugMask=snmp,ping,discovery,topology
```

E.9.4 /etc/netview/nv390mibs.def

```
#*********************************************************************
# Licensed Materials - Property of Tivoli Systems                    *
# 5697-B82 (c) Copyright Tivoli Systems 1999                         *
# All rights reserved.                                               *
#                                                                    *
# US Government Users Restricted Rights - Use, duplication or        *
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  *
#*********************************************************************
#*******************************************************************
#**In this file # in first column makes it a comment line.        ***
#*******************************************************************
/usr/lpp/netview/mibs/rfc1213-MIB-II.mib
/usr/lpp/netview/mibs/rfc1903.mib
/etc/netview/tcpip/dpiSimpl.mib
/etc/netview/tcpip/rfc1592b1.mib
/etc/netview/tcpip/ibm3172.mib
/etc/netview/tcpip/saMIB.mib
/etc/netview/tcpip/ibm3172b.mib
/etc/netview/tcpip/saMIB2.mib
/etc/netview/tcpip/mvstcpip.mib
/etc/netview/tcpip/mvstcpip2.mib
/etc/netview/tcpip/rfc1592b.mib
```

E.9.5 /etc/netview/nv390srvr.conf

```
************************************************************************
* Licensed Materials - Property of Tivoli Systems                      *
* 5697-B82 (c) Copyright Tivoli Systems 1999                           *
* All rights reserved.                                                 *
*                                                                      *
* US Government Users Restricted Rights - Use, duplication or          *
* disclosure restricted by GSA ADP Schedule Contract with IBM Corp.    *
************************************************************************

************************************************************************
***In this file * in first column makes it a comment line.          ***
***All keywords like snmphost must be lower case.                   ***
************************************************************************
************************************************************************
***Step one                                                         ***
**********                                                         ***
***   Copy this file to /etc/netview.  Change srvhostname,          ***
***   mibhost and snmphost with proper IP address or fully          ***
***   qualified hostname. If you wish to bring up snmpservice       ***
***   on a different port change srvport and snmpport. If you       ***
***   wish to to bring up mibservice on a different port change     ***
***   mibport.                                                      ***
***Step two                                                         ***
**********                                                         ***
***  This updated file from etc/netview/nv390srvr.conf             ***
***  needs to be copied to all the NMC server so that Mib Browser  ***
***  clients can connect to USS services.                          ***
***  The file will need to be copied to the following directory    ***
***  on the NMC server with the following file name:               ***
***  $(bindir)/TDS/server/db/current/settings/defaultmibbr.properties
***                                                                 ***
************************************************************************
srvhostname = 9.24.104.113
srvport = 4099
custdelimiter = *
mibhost = 9.24.104.113
mibport = 2099
snmphost = 9.24.104.113
snmpport = 2099
* height = 640
* width = 480
```

E.9.6 /etc/netview/snmp.conf

```
#***********************************************************************
# Licensed Materials - Property of Tivoli Systems                      *
# 5697-B82 (c) Copyright Tivoli Systems 1999                           *
# All rights reserved.                                                 *
#                                                                      *
# US Government Users Restricted Rights - Use, duplication or          *
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.    *
#***********************************************************************
#In this file # or blank character in the first column makes whole
#line a comment line.
#In this file * is used as a wild card character for Ip Address.
#A record consists of following Fields:
```

```
#ipRange:pingTimeout:pingRetry:statusPollInterval:getCommunityName:snmpTimeout
:snmpRetry:ConfigPollinterval:setCommunityName:remotePort:proxy
#You can create entries for target SNMP agents which will be used by snmpsrvc
for SNMP querry(if community string is not given) or NetView IP Disccovery
#In the first field you can code an IP address or range of IP address
#e.g. you can code 69.200.*.* or 69.200.40.20-40
#*********************************************************************

9.24.104.113:3s:3:30m:publicv1:5s:1:1d:publicv1:161:*
9.24.104.42:3s:3:30m:publicv1:5s:1:1d:publicv1:161:*
9.24.104.149:3s:3:30m:publiv1:5s:1:1d:publicv1:161:*
9.24.105.126:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
9.24.105.73:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.250.3:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.252.28:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
172.16.232.39:3s:3:30m:publicv1:2s:1:1d:publicv1:161:*
9.24.105.*:3s:3:30m:public:2s:1:1d:public:161:*
*.*.*.*:3s:3:30m:public:2s:1:1d:public:161:*
```

E.10 NETVIEW subsystem interface configuration files

E.10.1 USER.PROCLIB(NETVS03)

```
//CNMPSSI PROC
//NETVIEW  EXEC PGM=CNMINIT,TIME=1440,REGION=1250K,
//         PARM=(4000,200,'','SYSTEM','PPI','*NOARM',
//          'ONE',300,0),DPRTY=(13,13)
//STEPLIB  DD   DSN=NETVIEW.V1R3M0.CNMLINK,DISP=SHR
```

E.11 UNIX server configuration files

E.11.1 NETVIEW.V1R3USER.RA03N.DSIPARM(CNMSJUNX)

```
//CNMEUNIX JOB ,'NETVIEW UNIX SERVER',MSGCLASS=R,REGION=0M
//        EXEC PGM=BPXBATCH,
//             PARM='PGM /usr/lpp/netview/bin/cnmeunix'
//STEPLIB   DD DISP=SHR,DSN=NETVIEW.V1R3USER.RA03N.USERLINK
//          DD DISP=SHR,DSN=REXX.V1R3M0.SEAGALT
//*
//STDOUT    DD PATH='/tmp/cnmeunix.stdout',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=SIRWXU
//*
//STDERR    DD PATH='/tmp/cnmeunix.stderr',
//             PATHOPTS=(OWRONLY,OCREAT,OTRUNC),
//             PATHMODE=SIRWXU
//*
//STDENV   DD DSN=NETVIEW.V1R3USER.TCPPARMS(AON03ENV),DISP=SHR
//*-------------------------------------------------------------------
//*
//STDOUT  EXEC PGM=IKJEFT01,COND=((256,LE),EVEN)
//SYSTSPRT  DD SYSOUT=*
//*
//FROMHFS   DD PATH='/tmp/cnmeunix.stdout',
//             PATHOPTS=(ORDONLY,OCREAT)
//*
```

```
//TOSYSOUT  DD SYSOUT=*,
//             RECFM=F,BLKSIZE=255
//*
//SYSTSIN   DD DATA
         OCOPY INDD(FROMHFS)    +
               OUTDD(TOSYSOUT)  +
               TEXT             +
               CONVERT(YES)     +
               PATHOPTS(USE)
/*
//*_____________________________________________________________________
//*
//STDERR  EXEC PGM=IKJEFT01,COND=((256,LE),EVEN)
//SYSTSPRT  DD SYSOUT=*
//*
//FROMHFS   DD PATH='/tmp/cnmeunix.stderr',
//             PATHOPTS=(ORDONLY,OCREAT)
//*
//TOSYSOUT  DD SYSOUT=*,
//             RECFM=F,BLKSIZE=255
//*
//SYSTSIN   DD DATA
         OCOPY INDD(FROMHFS)    +
               OUTDD(TOSYSOUT)  +
               TEXT             +
               CONVERT(YES)     +
               PATHOPTS(USE)
/*
//
```

E.11.2 NETVIEW.V1R3USER.TCPPARMS(AON03ENV)

```
RESOLVER_CONFIG=//'TCPIP.TCPPARMS(TCPD03A)'
PATH=/bin:/usr/lpp/netview/bin:/usr/lpp/tcpip/bin:/usr/lpp/java/J1.1/bin:/etc/netview
MANPATH=/usr/lpp/netview/man/%L:/usr/man/%L
CLASSPATH=/usr/lpp/netview/lib/ihssnmps.jar:/usr/lpp/java/J1.1/lib/classes.zip:/usr/lpp/netv
iew/lib/ipdiscovery.jar
LIBPATH=/usr/lpp/netview/lib
LD_LIBRARY_PATH=/usr/lpp/netview/lib
JAVA_COMPILER=off
TZ=EST5EDT
```

Appendix F. Special notices

This publication is intended to help customers who want to manage TCP/IP applications running on an OS/390 system or monitor TCP/IP resources in a network from an OS/390 system using Simple Network Management Protocol (SNMP). The information in this publication is not intended as the specification of any programming interfaces that are provided by Communications Server for OS/390 IP Services. See the PUBLICATIONS section of the IBM Programming Announcement for CS for OS/390 IP for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating

environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	APPN
AS/400	CICS
DB2	DPI
IBM	MVS/ESA
Netfinity	OpenEdition
OS/2	OS/2 WARP
OS/390	PROFS
RACF	RMF
RS/6000	S/390
SecureWay	SP
SP1	System/390
VM/ESA	VTAM

The following terms are trademarks of other companies:

Tivoli, Manage. Anything. Anywhere.,The Power To Manage., Anything. Anywhere.,TME, NetView, Cross-Site, Tivoli Ready, Tivoli Certified, Planet Tivoli, and Tivoli Enterprise are trademarks or registered trademarks of Tivoli Systems Inc., an IBM company, in the United States, other countries, or both. In Denmark, Tivoli is a trademark licensed from Kjøbenhavns Sommer - Tivoli A/S.

C-bus is a trademark of Corollary, Inc. in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

PC Direct is a trademark of Ziff Communications Company in the United States and/or other countries and is used by IBM Corporation under license.

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through The Open Group.

SET and the SET logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.

Appendix G. Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

G.1 IBM Redbooks publications

For information on ordering these publications see “How to get IBM Redbooks” on page 321.

- *OS/390 eNetwork Communications Server TCP/IP Implementation Guide Volume 1: Configuration and Routing,* SG24-5227
- *OS/390 eNetwork Communications Server TCP/IP Implementation Guide Volume 2: UNIX Applications,* SG24-5228
- *OS/390 eNetwork Communications Server TCP/IP Implementation Guide Volume 3: MVS Applications,* SG24-5229
- *TCP/IP Tutorial and Technical Overview*, GG24-3376
- *An Introduction to Tivoli NetView for OS/390 V1R2*, SG24-5224
- *OS/390 Version 2 Release 6 UNIX System Services Implementation and Customization*, SG24-5178
- *Understanding LDAP*, SG24-4986
- *LDAP Implementation Cookbook*, SG24-5110
- *Networked Applications on OS/390 UNIX*, SG24-5447
- *RACF Support for Open Systems Technical Presentation,* GG26-2005
- *IP Network Design Guide,* SG24-2580

G.2 IBM Redbooks collections

Redbooks are also available on the following CD-ROMs. Click the CD-ROMs button at http://www.redbooks.ibm.com/ for information about all the CD-ROMs offered, updates and formats.

CD-ROM Title	Collection Kit Number
System/390 Redbooks Collection	SK2T-2177
Networking and Systems Management Redbooks Collection	SK2T-6022
Transaction Processing and Data Management Redbooks Collection	SK2T-8038
Lotus Redbooks Collection	SK2T-8039
Tivoli Redbooks Collection	SK2T-8044
AS/400 Redbooks Collection	SK2T-2849
Netfinity Hardware and Software Redbooks Collection	SK2T-8046
RS/6000 Redbooks Collection (BkMgr Format)	SK2T-8040
RS/6000 Redbooks Collection (PDF Format)	SK2T-8043
Application Development Redbooks Collection	SK2T-8037
IBM Enterprise Storage and Systems Management Solutions	SK3T-3694

G.3 Other resources

These publications are also relevant as further information sources:

- *OS/390 SecureWay Communications Server IP Migration*, SC31-8512

- *OS/390 SecureWay Communications Server IP Configuration*, GC31-8513
- *OS/390 SecureWay Communications Server IP User's Guide*, GC31-8514
- *OS/390 SecureWay Communications Server IP Programmer's Reference*, SC31-8515
- *OS/390 SecureWay Communications Server IP Application Programming Interface Guide*, SC31-8516
- *OS/390 SecureWay Communications Server IP Messages: Volume 1*, SC31-8517
- *OS/390 SecureWay Communications Server IP Messages: Volume 2*, SC31-8570
- *OS/390 SecureWay Communications Server IP Messages: Volume 3*, SC31-8674
- *OS/390 SecureWay Communications Server IP and SNA Codes*, SC31-8571
- *OS/390 SecureWay Communications Server IP Diagnosis Guide*, SC31-8521
- *IBM TCP/IP Performance Tuning Guide,* SC31-7188
- *Tivoli NetView for OS/390 Planning Guide Version 1 Release 3*, GC31-8226
- *Tivoli NetView for OS/390 Installation and Administration Guide Version 1 Release 3*, SC31-8236
- *Tivoli NetView for OS/390 Administration Reference Version 1 Release 3*, SC31-8222
- *Tivoli NetView for OS/390 User's Guide Version 1 Release 3*, GC31-8241
- *Tivoli NetView for OS/390 Automated Operations Network Customization Guide Version 1 Release 3*, SC31-8662
- *Tivoli NetView for OS/390 MultiSystem Manager User's Guide Version 1 Release 3*, GC31-8607
- *Tivoli NetView for OS/390 NetView Management Console User's Guide Version 1 Release 3*, GC31-8665
- *Tivoli NetView for OS/390 Customization: Using REXX and the NetView Command List Language Version 1 Release 3*, SC31-8231
- *Tivoli NetView for OS/390 Security Reference Version 1 Release 3*, SC31-8606
- *Tivoli NetView for OS/390 Resource Object Data Manager and GMFHS Programmer's Guide Version 1 Release 3*, SC31-8233
- *Tivoli NetView for OS/390 Customization: Using Pipes Version 1 Release 3*, SC31-8248
- *Tivoli NetView for OS/390 Command Reference Volume 1 Version 1 Release 3*, SC31-8227
- *Tivoli NetView for OS/390 Command Reference Volume 2 Version 1 Release 3*, SC31-8735
- *Tivoli NetView for OS/390 Messages and Codes Version 1 Release 3*, SC31-8237
- *OS/390 UNIX System Services Planning*, SC28-1890
- *OS/390 UNIX System Services User's Guide*, SC28-1891

- *OS/390 UNIX System Services Command Reference*, SC28-1892 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- *OS/390 UNIX System Services Messages and Codes,* SC28-1908
- *OS/390 Initialization and Tuning Reference*, SC28-1752 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- *OS/390 MVS System Messages, Vol. 1 (ABA-ASA),* GC28-1784 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- *OS/390 MVS System Messages, Vol. 2 (ASB-EZM),* GC28-1785 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- *OS/390 MVS System Messages, Vol. 3 (GDE-IEB),* GC28-1786 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- *OS/390 MVS Sy*stem Messages, Vol. 4 (IEC-IFD), GC28-1787 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- OS/390 MVS System Messages, Vol. 5 (IGD-IZP), GC28-1788 (available on CD-ROMs SK2T-6700 and SK2T-6718)
- *OS/390 Security Server (RACF) Command Language Reference*, SC28-1919
- *David Zeltserman, A Practical Guide to SNMPv3 and Network Management,* published by Prentice Hall, SR23-9114, ISBN 0-13-021453-1
- *W. Richard Stevens, TCP/IP Illustrated Volume 1 - The Protocols,* published by Addison-Wesley, SR28-5586, ISBN 0-201-63346-9

G.4 Referenced Web sites

These Web sites are also relevant as further information sources:

- `http://www.s390.ibm.com/` - IBM S/390 homepage
- `http://www.s390.ibm.com/networking/` - S/390 networking homepage
- `http://www.tivoli.com/nv390/` - Tivoli NetView for OS/390 homepage
- `http://www.tivoli.com/nv390_supported/` - Tivoli NetView for OS/390 supported functions homepage
- `http://www.s390.ibm.com/products/oe/bpxa1tun.html` - OS/390 UNIX System Services performance homepage

How to get IBM Redbooks

This section explains how both customers and IBM employees can find out about IBM Redbooks, redpieces, and CD-ROMs. A form for ordering books and CD-ROMs by fax or e-mail is also provided.

- **Redbooks Web Site** http://www.redbooks.ibm.com/

  Search for, view, download, or order hardcopy/CD-ROM Redbooks from the Redbooks Web site. Also read redpieces and download additional materials (code samples or diskette/CD-ROM images) from this Redbooks site.

  Redpieces are Redbooks in progress; not all Redbooks become redpieces and sometimes just a few chapters will be published this way. The intent is to get the information out much quicker than the formal publishing process allows.

- **E-mail Orders**

  Send orders by e-mail including information from the IBM Redbooks fax order form to:

	e-mail address
In United States	usib6fpl@ibmmail.com
Outside North America	Contact information is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Telephone Orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	Country coordinator phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Fax Orders**

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	Fax phone number is in the “How to Order” section at this site: http://www.elink.ibmlink.ibm.com/pbl/pbl

This information was current at the time of publication, but is continually subject to change. The latest information may be found at the Redbooks Web site.

IBM Intranet for Employees

IBM employees may register for information on workshops, residencies, and Redbooks by accessing the IBM Intranet Web site at http://w3.itso.ibm.com/ and clicking the ITSO Mailing List button. Look in the Materials repository for workshops, presentations, papers, and Web pages developed and written by the ITSO technical professionals; click the Additional Materials button. Employees may access MyNews at http://w3.ibm.com/ for redbook, residency, and workshop announcements.

IBM Redbooks fax order form

Please send me the following:

Title	Order Number	Quantity

First name	Last name	
Company		
Address		
City	Postal code	Country
Telephone number	Telefax number	VAT number

☐ Invoice to customer number

☐ Credit card number

Credit card expiration date	Card issued to	Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

Abbreviations and acronyms

APAR	authorized program analysis report
API	application programming interface
AON	Automated Operations Network
ATM	Asynchronous Transfer Mode
DDF	distributed data facility
DES	Data Encryption Standard
DFSMS	Data Facility Storage Management Subsystem
DHCP	Dynamic Host Configuration Protocol
DLL	dynamic link library
DPI	Distributed Protocol Interface
DSCP	differentiated services codepoint
EGP	External Gateway Protocol
ESP	Encapsulating Security Payload
GMFHS	Graphic Monitor Facility host subsystem
GUI	graphical user interface
HFS	Hierarchical File System
HMAC	Hashed-Based Message Authentication Code
HSRP	Hot Standby Router Protocol
IANA	Internet Assigned Numbers Authority
IBM	International Business Machines Corporation
ICMP	Internet control message protocol
IETF	Internet Engineering Task Force
I/O	input/output
IP	Internet Protocol
IS	integrated service
ITSO	International Technical Support Organization
IUCV	inter-user communication vehicle
LAN	local area network
LCS	LAN channel station
LDAP	Lightweight Directory Access Protocol
LFS	logical file system
LU	logical unit
MAE	Multiaccess Enclosure
MCS	multiple console support
MIB	Management Information Base
MPC	multipath channel
NGMF	NetView Graphic Monitor Facility
NMC	NetView management console
OSA	Open Systems Adapter
OSPF	Open Shortest Path First
PDU	protocol data units
PFS	physical file system
POSIX	portable operating system interface for computer environments
PU	physical unit
QDIO	Queued Direct I/O
RACF	Resource Access Control Facility
RFC	Request for Comments
RODM	Resource Object Data Manager
RSVP	Resource Reservation Protocol
RTT	round trip time
SGMP	Simple Gateway Monitoring Protocol
SLA	service level agreement
SLAPM	service level agreement performance monitor
SMI	Structure of Management Information
SNA	Systems Network Architecture
SNMP	Simple Network Management Protocol
SPUFI	SQL Processor Using File Input
SQE	SNMP Query Engine
SSI	MVS Subsystem Interface

TCB	Transmission Control Block
TCP	Transmission Control Protocol
TNF	termination notification facility
TTL	time to live
TOS	type of service
UDP	User Datagram Protocol
USM	user-based security model
VIPA	Virtual IP address
VMCF	virtual machine communication facility
XCF	cross-system coupling facility

Index

Symbols

A

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

IBM Redbooks review

Your feedback is valued by the Redbook authors. In particular we are interested in situations where a Redbook "made the difference" in a task or problem you encountered. Using one of the following methods, **please review the Redbook, addressing value, subject matter, structure, depth and quality as appropriate.**

- Use the online **Contact us** review redbook form found at **ibm.com**/redbooks
- Fax this form to: USA International Access Code + 1 914 432 8264
- Send your comments in an Internet note to redbook@us.ibm.com

Document Number **Redbook Title**	SG24-5866-00 Managing OS/390 TCP/IP with SNMP
Review	
What other subjects would you like to see IBM Redbooks address?	
Please rate your overall satisfaction:	O Very Good O Good O Average O Poor
Please identify yourself as belonging to one of the following groups:	O Customer O Business Partner O Solution Developer O IBM, Lotus or Tivoli Employee O None of the above
Your email address: The data you provide here may be used to provide you with information from IBM or our business partners about our products, services or activities.	O Please do not use the information collected here for future marketing or promotional contacts or other communications beyond the scope of this transaction.
Questions about IBM's privacy policy?	The following link explains how we protect your personal information. **ibm.com**/privacy/yourprivacy/

Managing OS/390 TCP/IP with SNMP

(0.5” spine)
0.5”<->0.875”
250 <-> 459 pages

Managing OS/390 TCP/IP with SNMP

Sample configuration for managing TCP/IP resources on OS/390

How to manage IP resources in your network from an OS/390 system

Take advantage of SNMPv3 supported by CS for OS/390

This book addresses the major networking issues facing many large enterprises today: how to manage IP resources on an OS/390 system and how to manage all IP resources in an IP network from an OS/390 system.

System Network Architecture (SNA) was the predominant network architecture for OS/390 systems and as the shift from this hierarchical structure to the heterogeneous peer-to-peer TCP/IP environment takes place, it becomes necessary for the network management infrastructure to adapt to the changing environment. The enterprise of today increasingly relies more on TCP/IP for both the network environment and applications. Effective and timely host management is essential for maintaining reliability and availability.

This redbook provides information that will help a network administrator to implement and configure the Simple Network Management Protocol (SNMP) function of Communications Server for OS/390 IP Services and Tivoli NetView for OS/390. The SNMP Version 3 (SNMPv3) function of CS for OS/390 provides a secure management environment that allows an OS/390 host to become a managed and/or managing node within the enterprise. Tivoli NetView for OS/390 provides a powerful, centralized interface for managing IP nodes regardless of which platform it is running on. In this redbook you will find many examples from the ITSO environment that will assist in managing OS/390 TCP/IP with SNMP.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by IBM's International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-5866-00 ISBN 0738417289