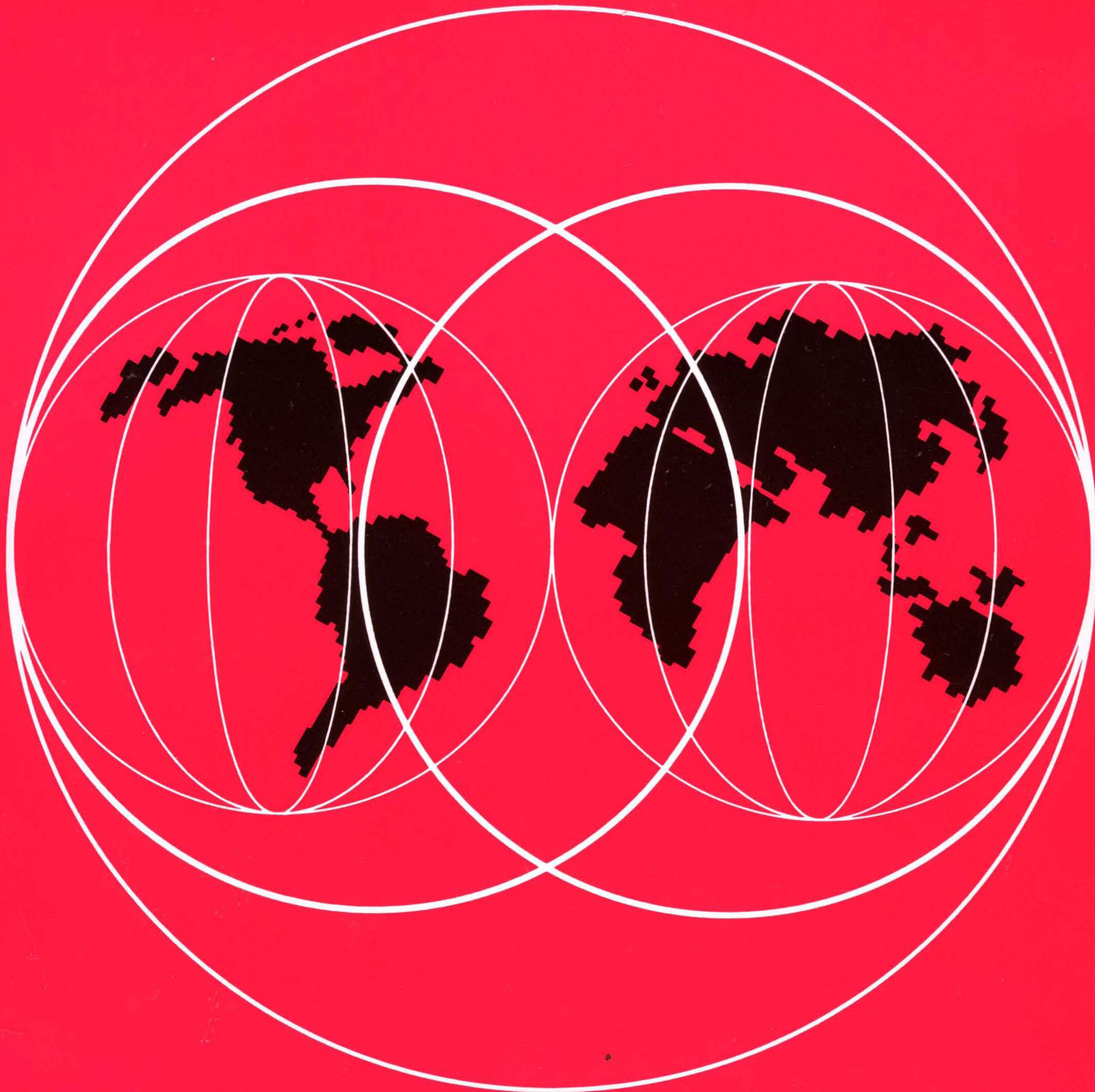


IBM

International Technical Support Centers

A Collection of IBM LAN NetView White Papers

GG24-4207-00

A Collection of IBM LAN NetView Papers

Document Number GG24-4207-00

November 1993

International Technical Support Organization
Austin Center

Take Note!

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

First Edition (November 1993)

This edition applies to Version 1, Release 0 of the IBM LAN NetView family of products.

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

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 948S, Building 821 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Abstract

This document is a collection of papers discussing the IBM LAN NetView family of products. These white papers include topics ranging from general information about the products themselves to more detailed information regarding the use and programming interfaces of the products.

This document will be of interest to customers and IBM system engineers who are evaluating IBM LAN NetView as a solution for distributed management of LAN-based systems.

PS (296 pages)

Contents

Figures

Special Notices

This publication is intended to help IBM system engineers and customers to gain insight into the IBM LAN NetView family of products. The information in this publication is not intended as the specification of any programming interfaces that are provided by the IBM LAN NetView products. See the PUBLICATIONS section of the IBM Programming Announcement for the IBM LAN NetView products 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 Commercial Relations, IBM Corporation, Purchase, NY 10577.

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.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

ADSTAR	AIX
AIX SystemView NetView/6000	AS/400
Communications Manager/2	datagLANce
DB2/2	Distributed Console Access Facility
HPFS	Hyperblock
IBM	LAN Distance
LAN NetView	LAN NetView Agents Extended
LAN NetView Agents for DOS	LAN NetView Enabler
LAN NetView Fix	LAN NetView Manage
LAN NetView Management Utilities for OS/2	LAN NetView Monitor
LAN NetView Scan	LAN NetView Start
LAN NetView Tie	LAN Network Manager
LAN Station Manager	LANDP

NTS/2
NetView/6000
OS/2 Extended Edition
OS/400
Personal System/2
Presentation Manager
RTIC
SAA
Save Utility/2
SystemView
Wide Area Connector
NetView
OS/2
OS/2 Extended Services
PS/2
Portmaster
RISC System/6000
S/390
SPM/2
System Performance Monitor/2
Theseus2
Workplace Shell

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

AlertVIEW	Shany Computers, LTD
Allerion	Allerion, Inc.
AppleTalk	Apple Computer, Inc.
BEST/1-Visualizer for OS/2, Performance Assurance, Analyze for OS/2	BGS Systems, Inc.
CompuServe	CompuServe Incorporated
DEC, OpenVMS, DECnet	Digital Equipment Corporation
Dolphin ESP	Dolphin Networks
Gradient Technologies, iFOR/LS	Gradient Technologies, Inc.
HP, OpenView	Hewlett-Packard Corporation
HiTecSoft	High Technology Software Corp.
IEEE	International Institute of Engineers
IMPACT	Allen Systems Group, Inc.
Intel	Intel Corporation
Legato Systems, NetWorker	Legato Systems, Inc.
Lotus	Lotus Corporation
MASS/2	Parallan Computer, Inc.
Microcom, LANlord	Microcom, Inc.
Microsoft, Windows, Windows NT	Microsoft Corporation
NDIS	3Com Corporation
NetWare, IPX, ODI	Novell Inc.
Opening Technologies, OPENT1	Opening Technologies, Inc.
Osrm2	C.O.L. Systems, Inc.
ProTools, Foundation Manager, Cornerstone Agent	ProTools, Inc.
Qualitas, BlueMAX, 386MAX	Qualitas, Inc.
Quarterdeck, QEMM-386	Quarterdeck Office Systems
Service Point/32	Strategic Solutions International Corp.
SunNet Manager	Sun Systems, Inc.
Tower	Pragma Systems, Inc.
UNIX	X/Open Compnay Ltd.
Universal Premises Network	Luxcom, Inc.
VINES	Banyan Systems, Inc.
X/Open Management Protocols	X/Open Company Limited
XNS, Ethernet	Xerox Corporation

Preface

This document is intended to provide an assortment of general and technical information regarding the IBM LAN NetView family of products. It contains a collection of papers written by many different authors involved in the development and release of LAN NetView.

This document is intended to supplement the marketing information available on LAN NetView. It will provide additional technical details to be used to evaluate the applicability of LAN NetView to a customer's distributed management requirements.

How This Document is Organized

The document is organized into four parts:

- Part 1 contains general information in the following papers:
  - Chapter 2, "OS/2 Distributed Systems Management: LAN NetView Family of Products" on page 5
  - Chapter 3, "LAN NetView: Framework Overview" on page 51
- Part 2 contains planning and usage information papers:
  - Chapter 4, "Planning a Management Scheme Using the IBM LAN NetView Family of Products" on page 75
  - Chapter 5, "LAN NetView Hardware Requirements and Performance" on page 83
  - Chapter 6, "Hints and Tips for Using Topology Management Services" on page 93
  - Chapter 7, "Comparison of LAN NetView Monitor 1.0 and SPM/2 2.0" on page 99
  - Chapter 8, "Using IBM LAN NetView Manage Version 1.0 and IBM LAN Network Manager Version 1.1 Together" on page 109
  - Chapter 9, "Remote LAN Management Using IBM LAN NetView" on page 117
- Part 3 contains papers pertaining to programming in a LAN NetView environment:
  - Chapter 10, "Introduction to the IBM LAN NetView Application Programming Interface" on page 135
  - Chapter 11, "Managing OS/2, DOS and Windows Stations from LAN NetView" on page 143
  - Chapter 12, "IBM LAN NetView: Agents and Objects Overview" on page 149
  - Chapter 13, "Using a LAN NetView Managed-Object Catalog" on page 195
  - Chapter 14, "Accessing SNMP Managed Objects Using CMIP Services from the LAN NetView XMP API" on page 207

- Chapter 15, "Providing SNMP Management Using IBM LAN NetView" on page 221
- Chapter 16, "Exploiting LAN NetView Metadata Services for a Generic Management Application" on page 227

- Part 4 contains information relating to products available from independent software vendors (ISVs) that work in conjunction with the LAN NetView family:
  - Chapter 17, "Opportunities for LAN NetView-based Applications" on page 255
  - Chapter 18, "LAN NetView Catalog of Value-added Offerings" on page 263

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *IBM LAN NetView Manage User Interface Developer's Guide*, S96F-8753-00
- *IBM LAN NetView Manage Administration Guide*, S96F-8492-00
- *IBM LAN NetView Manage and Enabler Developer's Guide*, S96F-8493-00
- *IBM LAN NetView Manage and Enabler Developer's Reference*, S96F-8494-00
- *IBM LAN NetView Operating Systems Agents Managed-Object Catalog*, S96F-8495-00
- *IBM LAN NetView Agents Extended Database Manager Managed-Object Catalog*, S96F-8496-00
- *IBM LAN NetView Agents Extended Communications Manager Managed-Object Catalog*, S96F-8497-00
- *IBM LAN NetView Agents Extended LAN Server Managed-Object Catalog*, S96F-8498-00
- *FFST/2 Administration Guide*, S96F-8593-00

International Technical Support Organization Publications

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

Bibliography of International Technical Support Organization Technical Bulletins, GG24-3070.

Acknowledgments

This document was compiled by:

Bart Jacob	International Technical Support Organization, Austin Center
Al Marcus	LAN Systems - IBM Austin

Thanks to the entire LAN NetView development team for their assistance and contributions to the papers included in this document.

Chapter 1. Introduction

Much of the information contained in this collection of papers is of general interest to anyone seeking information on the IBM* LAN NetView* family of products. However, some topics will be of specific interest to programmers, network administrators, Independent Software Vendors (ISVs), marketing, or customers. With that in mind, we've grouped the papers into four categories. They are: General Information, Systems Administration, Programming Information, and ISV-Related Products and Information. Since the General Information section includes overviews on the LAN NetView family of products, previous LAN NetView knowledge is not required. However, there are prerequisites in some cases in the programming sections in terms of object-oriented concepts and use of standard programming interfaces.

Disclaimer

This information is not intended to be an assertion of future action. IBM expressly reserves the right to change or withdraw current products that may or may not have the same characteristics or codes listed in this document. It is possible that this material may contain reference to, or information about, IBM products that are not announced in your country.

IBM has not evaluated the listed applications of other companies, and do not recommend or endorse directly to verify product function, prerequisites, availability and suitability for the users′ purposes.

This document is based on the most recent information provided by the companies listed as the sources for the applications at the time of printing this document.

Trademark attributions for the listed applications are based on the information provided by each company.

This document may contain inaccuracies or typographical errors. IBM may change the contents without notice.

Part 1. General Information

Chapter 2. OS/2 Distributed Systems Management: LAN NetView Family of Products

The purpose of this paper is to address the following questions:

- Who are the LAN NetView product's customers?
- What product content is being delivered by the initial products?
- What kind of problems can be solved by these products?
- Which of the LAN NetView products are necessary on each workstation?
- How does the LAN NetView product fit with other products a customer may have in place?
- What kinds of additions to the LAN NetView family can be expected in the near future?

2.1 LAN NetView Environments and Users

It may be useful to differentiate between two types of LAN environments that the LAN NetView product is designed to manage:

- Those networks for which the workstation is the largest system unit on the LAN (that is, no host system present)
- Those LAN networks which have a host attached.

The first group (no host) will be referred to as the Q2 environment and the second group (with host) will be referred to as Q4 for future reference, following the "Four Quadrants" Opportunity Structure model.

2.1.1 Q2 Environment - No Host

A large portion of the Q2 customers are the small to medium size enterprises. Their workgroup LANs are typically purchased through dealers or value added remarketers (VARs) who market turnkey packages to this customer set. These customers will typically not have support expertise in the organization and will rely on the dealer/VAR for ongoing support.

The environment for workgroup LANs includes a set of user workstations and one or more servers. The servers are primarily for file sharing, printer sharing, and electronic mail. The applications run on the LAN are normally vendor applications which are purchased as a package with the LAN hardware. Any systems management function installed on the LAN would ordinarily be recommended by the dealer/vendor. The minimum support would include capability to detect hardware errors and security violations. Configuration management function would be present when the dealer/vendor needed the function for initial installation. The dealer/vendor may also use a performance manager as a part of tuning and ongoing support. The day-to-day management of the customer workgroup LAN may be done remotely by the dealer or a service provider. This remote support would be via telephone connection to access the LAN management function.

A second class of Q2 customers are departments in large enterprises. In many cases they are just as dependent on a dealer/vendor as the small to medium size customer. The department may have a unique need for personal systems applications not associated with the enterprise host environment. The systems management application usage would be the same as that for the other customers who depend on a dealer/vendor for support. Within the large customer set there is a second group of non-host connected LAN users who are more sophisticated and do their own selection of hardware and software and provide the bulk of their own support. These would include technical departments doing mathematical kinds of applications, or specialty departments whose applications do not lend themselves to host processing. This group would again require problem management capability and potentially the configuration and performance applications.

To summarize, the Q2 requirements for systems management applications include problem, performance, and configuration management. The management console supporting the LAN may either be local or at the dealers'/vendors' or service representatives' site.

The LAN sizes in this group would be in the two to twenty-five workstations range and would include both standalone and bridged LANs.

2.1.2 Q4 Environment - Host Present

There are two types of LAN configurations present in the Q4 environment that are of interest. The first type is the bridged LAN. A large percentage of Q4 LANs fall into this category. There is a host connection as well as a connection between LANs which is not host associated. The second type is simply host-attached (no separate bridge attachment). There are a large number of this type of Q4 LANs as well, but not as many as the bridged type. Both of these environments require systems and network management and the same applications set would apply to both.

The Q4 environment is made up mostly of medium to large accounts. Since the Q4 customers have a host system, the systems management function may tend to be centralized at the host location. Where the administration and operation of the LAN is centralized, the ability to provide management remotely is needed. Where distributed systems management is used there is usually some management connection to the central host as well.

2.1.3 User Descriptions

2.1.3.1 LAN Site

The primary user of the LAN NetView products at the LAN site is the system administrator. The LAN System Administrator would generally have the following duties associated with the installation and operation of the LAN:

- Coordination of the installation of hardware and software for the LAN. This includes the receiving of the components, scheduling of installers, system setup, user training, connection to host or other services, and performance tuning.
- Maintenance of servers, including backup/restore, security, issuance of passwords, problem determination for server errors, and monitoring basic resource usage.
- Interface with the service provider(s) for problem reporting, additions to the LAN, and scheduling preventative maintenance.

- End-user support. This support includes answering questions about the operation of application and operating system software, trouble reports, move requests, adding/deleting users from the system, password control, etc.

In many instances the system administrator function will be fulfilled by the dealer/vendor, with the customer providing only a person to make the initial contact.

A secondary user class on the LAN site would be technicians who are very knowledgeable of personal systems and LANs, and wish to be self- sufficient. In this environment there may be multiple users with access to tools and be able to modify workstation or even server environments on the LAN. Examples are user groups with LANs used only for mail/ messaging between engineers, or used only for printer sharing.

2.1.3.2 Central Site

There are many possible users of systems management tools at a central site for a large customer. All these users can be expected to be technicians with expertise in one or more areas. The following list describes some of these users:

- Help Desk - The personnel operating the help desk take the customer calls for assistance. The calls may involve hardware problems, software problems, or both. The help desk will normally try to do the initial problem determination, logging, and provide solutions when possible, or route the problem to other technical support levels.
- Installation - The installation group is normally responsible for the rollout of new LANs, hardware and software upgrades to existing LANs including the scheduling of hardware/software for moves, and additions of new users to the LAN.
- LAN Operations - The LAN Ops group is responsible for the monitoring of daily operations of the LANs. They become involved in problem resolution and the elimination of components that are having an adverse affect on the operation of the LAN or network of LANs.
- Network Operations - The management of the Wide Area Network (WAN) is the responsibility of this group. In some cases they may also manage the LAN. They provide similar functions for the WAN as the LAN Ops group does for the LAN.
- Security Administration - The managing of security on a network includes authorizing and revoking access authority to users of the network, monitoring unauthorized access attempts, and assisting in resolution of problem situations where security plays a role.
- Systems Programmer - The management of the operating system environment is performed by the systems programmers. In many cases they are also responsible for purchased application code.
- Application Programmer - Responsible for the creation and maintenance of applications used on the LAN.
- Performance Analyst - This specialist works with systems programmers to determine the best way to configure the network and systems to provide the desired level of performance. The system may require tuning for overall workload throughput or for critical response times for example. The analyst

also monitors the network performance with the aim of eliminating performance bottlenecks before they become problems.

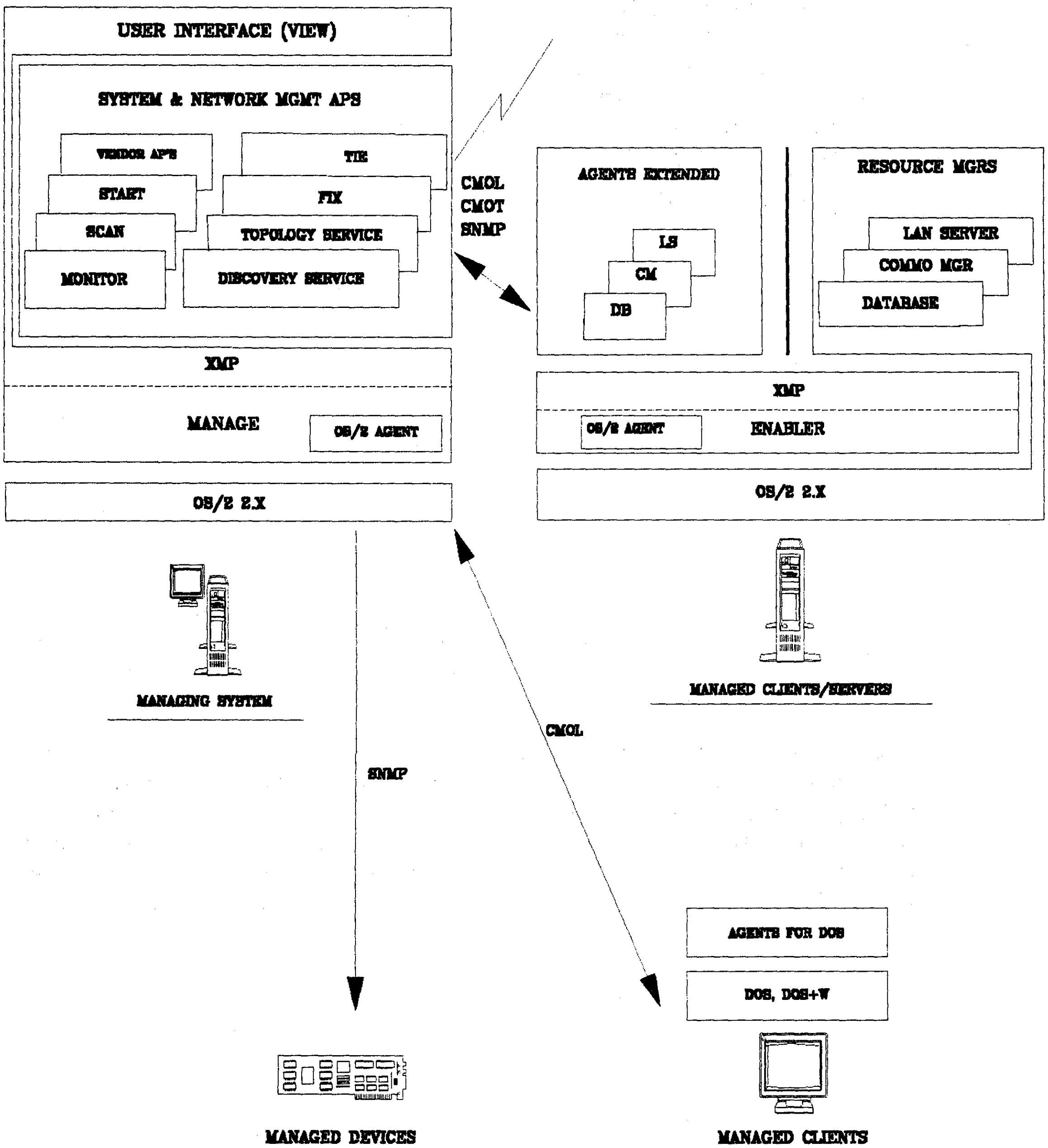
- Capacity Planner - There are many resources whose capacity levels require monitoring to maintain desired levels of end-user service. It is the role of the capacity planner to project when these capacities will be exceeded and take the appropriate steps to resolve the pending problem by adding additional resource, restricting its usage, or otherwise resolving the contention for the resource. Common capacities requiring attention by capacity planners include:
  - LAN Server - DASD, Memory, Printers (Queues, Spools, etc.)
  - Database Server - Transactions, Memory, DASD, Directories
  - Communications Server - Response Time, Memory, Line Use, Error rates
  - LAN Adapter usage
  - Bridge Traffic
- Vendor Support - In order to provide the necessary data to fix a problem, vendor support personnel may need access to systems management data.

2.2 LAN NetView Family of Products

The implementation of a comprehensive systems management solution for OS/2 LANs requires the implementation of a "managing system" on the OS/2 2.x platform and managed system services and resource agents for each "managed system" type and resource to be supported. The LAN NetView family of products was conceived to provide this comprehensive solution.

The concept was to provide a structure built of open-architected elements and interfaces that would allow others to contribute to the solution, with their own systems management applications to complement the ones IBM is providing. Others may wish to develop agents for their resources that would allow them to be managed by the LAN NetView product. The diagram shows a high-level view of the LAN NetView systems management structure and managed resources.

LAN NetView



As part of the development process in creating the LAN NetView product set, IBM initiated a customer evaluation program for the family of products beginning with the framework products in 4Q92 and including applications in early 1993. Product content, packaging, and general availability decisions have been influenced as a result of these customer evaluations. The resulting product set, which represents the initial offering of the LAN NetView family of products, will first be described at an executive overview level followed by more detailed descriptions.

2.2.1 LAN NetView Framework Overview

The IBM LAN NetView family of products includes a set of products that form the strategic framework upon which the systems management functions can be built and therefore lay the foundation for the solution. These framework products provide the common infrastructure, services, and support elements that:

- Create the "managing system" environment on OS/2 2.x for management applications to be built upon
- Create the "managed system" environment on systems with OS/2 2.x, IBM DOS 5.0/6.1, Microsoft** DOS 5.0/6.0, and DOS 5.0/6.0/6.1 with Microsoft Windows** 3.0 or 3.1 installed, and which allow resource agents to manage the system resources
- Provide the resource agents required to manage not only the operating system resources for the supported systems, but also the OS/2 subsystem resources for LAN Server 3.0, Communications Manager/2*, Extended Services* 1.0 Database Manager, and DB2/2*.

Through use of these framework services, the task of creating systems management applications and resource agents is greatly simplified. Developers can focus their attention on delivering the specific function intended, and not in duplicating these common elements. Greater consistency and interoperability between applications and agents are also achieved. Among the common elements provided by the set of framework products are common user interface services, communications services supporting multiple protocols and transports, event and metadata services, common programming interfaces, and discovery and topology services.

The LAN NetView family of products include the following framework products:

- LAN NetView Manage*: the common services that form the framework to build management applications upon. These common services include:
  - X/Open APIs for data manipulation and management protocols (XOM & XMP)
  - Management protocols supported include CMIP and SNMP (SNMP devices such as bridges and routers are able to be managed in addition to the CMIP-based resource agents provided)
  - Topology/Discovery services to determine and depict the relationships between the various system and network resources along with their status. Since the system environment that the LAN NetView product is intended to manage is composed of a collection of LAN connected workstations, it is useful to build and maintain topological information about the objects on the network; what objects exist, where are they, and how do they interrelate?

    Topology Services allow for the building and maintaining of topological information necessary to create topological views of the LAN NetView managed objects, and to display topology maps through use of the View presentation services. Topology information is maintained separately from the user interface services to facilitate sharing of the topology information.

    Discovery Services allow for the gathering of topological information by initiating network searches. The results from Discovery searches can be used by Topology Services to prime the topology information base and to maintain the integrity of it's topological information.

- Event Management Services logs event data and routes event reports to the appropriate management programs
- Metadata Services store and retrieve managed object class definitions
- Process Control Services coordinate the startup and shutdown of management programs
- A Mini-application named Request Manager can be used to display SNMP device and/or workstation resource attributes/values using supported Management Information Bases (MIBs)
- A graphical user interface component, called View, is included for displaying managed resources and interfacing with the LAN NetView systems management application functions supporting those resources. The common user interface conforms to SystemView Integration Level 2 and the CUA-91 (Common User Access-91) specification. A set of enabling services allow easy extension of the presentation metaphor to include functions provided by systems management applications. These services are made available via the System Object Model (SOM) programming interface of OS/2.

- LAN NetView Enabler*: managed system services are provided for OS/2. These services are a subset of those provided on the managing system and allow the system's resources to be managed by the applications on the managing system. Enabler also includes the CMIP-based agents for OS/2 2.x, IBM LAN Requester V3.0, and LAN NetView Monitor that would be required by OS/2 managed systems. Agents allow the managing system to request data about the resource managers, and manage the resource manager by setting and changing values.
- LAN NetView Agents Extended*: CMIP-based agents are provided for managing the OS/2 subsystem resources belonging to the Communications Manager/2, Database Manager or DB2/2, and LAN Server subsystems.
- LAN NetView Agents for DOS*: CMIP-based agents are also provided for DOS 5.0 (IBM and Microsoft), DOS 6.1 (IBM), DOS 6.0 (Microsoft), DOS 5.0/6.0/6.1 with Windows 3.0 or 3.1 that allow their operating system resources to be managed.

IBM is encouraging vendors to provide agents for their system resources to allow the LAN NetView product to manage those resources. IBM is providing technical assistance to ISVs in support of these efforts.

2.2.2 LAN NetView Applications Overview

IBM is also providing systems management applications in conjunction with the managing system framework, LAN NetView Manage. These systems management applications include:

- LAN NetView Monitor*: system performance management for DASD/RAM/Processor monitoring of the OS/2 workstations and servers.

  Based on the System Performance Monitor/2* (SPM/2*) technology, IBM is providing an IBM SystemView-compliant performance management solution for the OS/2 2.x operating system environment which runs on LAN NetView Manage. This solution will support the collection and analysis of a rich set of metrics for OS/2 2.x performance-critical resources. In addition to the base operating system, metrics from IBM system extensions such as the LAN

Server/Requester, Communications Manager and Database Manager will also be supported.

Several functions, all written to the XMP API, will be provided to assist in graphing, recording, reporting and analyzing performance data collected in a SQL database. These functions will work in tandem with the OS/2 2.x resource manager agents contained in the LAN NetView Enabler product and the LAN NetView Agents Extended product, to satisfy performance management requirements such as threshold detection, alert generation and data summarization.

This performance solution will enable system administrators to monitor system performance, analyze performance trends/problems and use this solution as an aid for performance tuning, load balancing, and for managing network growth in an OS/2 LAN environment. Application developers will be able to better analyze designs, verify performance objectives, and optimize application performance for the OS/2 2.x environment. Furthermore, this solution will aid capacity planners in benchmarking and performance modeling exercises.

- LAN NetView Fix*: system problem management for reporting hardware and software failures with some automated recovery.

  Systems management tools that aid in problem management are a key source of controlling management costs. The LAN NetView Fix product will assist customers in managing Information Systems (I/S) networks either locally or remotely, and in providing increased level of service to the entire I/S user community. The increased level of service is the result of coordination and automation of error recovery, as well as integration with other LAN NetView product functions. Service is further enhanced by the ability to commonly manage OS/2, DOS, and OEM systems from a single managing workstation on a LAN.

  The LAN NetView Fix product provides a focal point and a control point for standalone or host-connected LANs. The LAN NetView Fix 1.0 function enables a system administrator to automate bypass or recovery procedures. Event management services provided by LAN NetView Manage 1.0 are used to filter and collect OSI events emitted by agents and FFST/2-enabled applications on the network. Further, user-modifiable filtering of input events is supported, as well as automated invocation of external procedures and management operations.

- LAN NetView Tie*: NetView gateway service for collecting and transforming OSI performance and fault alarms for transmission to NetView.

  The CMIP management protocols used by the LAN NetView applications and agents are different from those used by the host-based NetView product family. The conversion of LAN-based CMIP protocols to the SNA/MS network management protocols used by NetView is accomplished by this product. As a result, alarms that require host involvement can be transmitted to NetView/MVS as network alerts for further processing. The OS/2 Communications Manager facilities will be used for the SNA transport to the NetView host.

- LAN NetView Scan*: system configuration to determine the current configuration of a workstation.

  This systems management tool collects and monitors OS/2 workstation configuration data (both software and hardware, including Vital Product Data). LAN NetView Scan uses the topology/discovery services of the LAN

NetView Manage product to gather pertinent configuration data about specific workstations and supports the configuration database or repository. File Transfer services are provided that support either TCP/IP or NetBIOS protocols and can be used to copy configuration files (for example, .INI files) from the managed workstations.

- LAN NetView Start*: system configuration planning and management for configuring OS/2 workstations and servers without end-user involvement.

  The LAN NetView Start product plays a key role in IBM's CID (Configuration/Installation/Distribution) strategy which enables remote, unattended installation of workstations on an OS/2 LAN. The LAN NetView Start program generates the configuration response files and command or REXX files for use with Network Transport Services/2* (NTS/2*) or NetView Distribution Manager/2* (NvDM/2*), that allow workstations requiring configuration changes to be remotely installed with automated program distribution. With LAN NetView Start 1.1's object-oriented graphical user interface, the administrator can quickly and easily construct or modify the network topology.

  The initial release of this product preceded delivery of LAN NetView Manage and therefore runs as a standalone product. It complements the framework and other applications of the LAN NetView family without being integrated with the framework. Future plans may warrant this integration.

These applications will be discussed in more detail in later sections. Over time other existing IBM applications will be ported to, integrated with, or launched by this LAN NetView framework environment as well. Launching of the LAN NetView Management Utilities* for OS/2 (LMU) product will be possible with the initial offering.

While these applications are being developed by IBM, others will be supplied by leading vendors of systems management products. IBM is actively soliciting specific applications as well as generally encouraging vendors to develop their applications on the LAN NetView framework, with documentation detailing what is involved in developing an application (or porting an existing application) to run on this framework as well as providing technical assistance. Examples of vendor-developed applications on the LAN NetView Manage framework are:

- The NetWare** Services Manager for LAN NetView product, from Novell, provides full management services for Novell 3.11 and 4.0 networks.
- LANlord**, from Microcom**, provides several systems management functions for OS/2, DOS, and DOS+Windows workstations.
- Protools**' Network Analysis Series product provides comprehensive LAN performance analysis on networks managed by the LAN NetView product.
- AlertVIEW** from Shany, Inc. provides an application sniffer function for troubleshooting application errors on LAN NetView managed systems.

More detailed descriptions of these applications are provided in the section entitled "LAN NetView Applications from Other Vendors".

Some vendor-developed LAN NetView applications will be marketed by both IBM and the vendor, others will only be marketed by the vendor. Some 15 vendors are presently preparing LAN NetView applications.

2.3 LAN NetView Framework Product Descriptions

The LAN NetView framework products are described in more detail in the sections that follow. Each individual product description is followed by a corresponding "Customer Value" description for that particular product. For the framework products, the value statements most directly apply to systems management application and agent developers since they will derive the direct benefit. All values apply as well to the end-user either directly or indirectly through the integrated applications. The products are described in the following order:

- LAN NetView Manage Version 1.0
- LAN NetView Enabler Version 1.0
- LAN NetView Agents for DOS Version 1.0
- LAN NetView Agents Extended Version 1.0

2.3.1 LAN NetView Manage Version 1.0 - Description

2.3.1.1 Overview

The LAN NetView Manage product contains the base services for integrating applications that manage LAN-based resources. The services include those needed by systems management applications to manage the resources, called the manage "platform" and a set of services that applications use to be integrated from the end-users point of view, called the manage "view" services. The terms "platform" and "view" will be used to distinguish between these sets of services where required in this product description.

The IBM LAN NetView Manage Version 1.0 platform is the base for the Managing System in the OSI management model and offers standards-based interfaces and services to allow applications to be written to manage OS/2, DOS, and OEM systems or devices. The primary programming interface for application and agent development is the X/Open Management Protocol** Application Programming Interface, or XMP API. This interface is also endorsed by IBM in the SystemView architecture and by the Open Software Foundation (OSF). Through this interface applications have transparent access to a variety of standard management and communication protocols as well as object registration, event, and messaging services.

The LAN NetView Manage view component provides the user interface for the LAN NetView Manage product services and systems management applications. Together, the view and platform portions of LAN NetView Manage 1.0 on an OS/2 managing system provide the basis for services that users require in order to manage LAN and WAN-connected workstations, servers, and hardware such as bridges and routers. Additional user level function is provided by IBM and ISV applications running in the LAN NetView Manage 1.0 environment.

2.3.1.2 Functional Description

The IBM LAN NetView Manage product provides the developer with open and well-documented APIs for application development. The APIs access a set of system services which allow the developer to create applications without the need to know communications protocols or partner application location. This ease of programming should greatly increase the programmer's productivity.

The managing system will communicate with the managed systems using one of two industry standard management protocols: CMIP and SNMP. IBM LAN

NetView Manage 1.0 is implemented using many IBM and industry standards including IBM SystemView, IBM SAA*, OSI CMIS/CMIP & MIM (GDMO), OSF Distributed Management Environment (DME), TCP/IP SNMP, IEEE** Heterogeneous LAN Management (HLM), and others.

Additional function is included in the IBM LAN NetView Manage product to discover the resources on the network and present them to the user. This discovery function will access the entire network (or user defined subset) and provide an icon on the display for each node, including client and server workstations and other nodes such as SNMP devices like bridges and routers. The discovery application will also keep the topology updated as resources are added or deleted from the network. The topology presentation may be in the form of a physical view of a LAN or via multiple logical views. The topology is displayed using the view component of IBM LAN NetView Manage 1.0. This provides the user with a user interface consistent with the IBM SystemView definitions.

The IBM LAN NetView Manage product is built using the Hewlett Packard** (HP**) OpenView** Network Management Server Product Version 3.1 as the base. The HP OpenView Network Management Server is an open application development environment, and is built on the HP OpenView architecture which was derived from the OSI Management Framework. This architecture has proven to be very strong in the industry with respect to standards conformance, heterogeneity, scalability, flexibility, extensibility, ease of use, MIB support, and protocol support.

The HP OpenView architecture provides the basis for a standards-based approach of treating the entire network as a collection of managed objects manipulated by applications and management services. This standards-based approach facilitates productivity improvements in application development and application portability.

IBM has made modifications to the HP product to enhance its performance in the OS/2 environment. The proprietary HP programming interface has been replaced with the XMP API. This API was jointly created by Group Bull and Hewlett-Packard with assistance from IBM. This API has also been endorsed by the OSF.

The management API provides access to a set of services for the applications and users which include:

- Discovery service to find attached systems and devices.
- Topology service to integrate the discovery data from the various protocols (TCP/IP, LLC, & IPX**) to maintain awareness of the currently managed systems.
- Object Registration services to register managed object locations and resolve object names from applications into the network address required.
- Communications services for access to agents using one of the industry-standard management protocols including:
  - Common Management Information Protocol (CMIP)
  - Simple Network Management Protocol (SNMP)

  The management protocols may be used over multiple transports. CMIP may be used over TCP/IP as CMOT, and over the LAN Logical Link Control as CMOL. SNMP may be used over UDP/IP. Both synchronous and asynchronous messaging is supported.

- Event Management services allow applications to register for selected notifications which may be emitted by the agents, and receive the notifications as they occur.
- Metadata services allow an application to read the structure of managed objects in GDMO form. These services allow an application to be written which learns about the objects it manages as the application is running. Such generic applications become more valuable as additional agents are added to the configuration along with their corresponding object structures.

The View component of LAN NetView Manage V1.0 provides developer support for systems management applications in the following areas:

- Application Access - Services are provided to manage the LAN NetView Manage V1.0 application workplace, and to invoke the appropriate application functions when actions are selected for displayed object icons.
- Navigation Services - LAN NetView Manage V1.0 navigation services enhance developer productivity, in that the coding effort to imbed navigation capability between application functions is reduced or eliminated.

This SOM-based topology display portion of the Manage product (called View) is the first level of integration of systems management functions from IBM and non-IBM sources and provides the user with a consistent view of the network resources and management tools. The resource view is based on information about the network resources gathered by the discovery service and integrated by the topology service. These were described earlier. Various layouts can be selected to give users the most informative view of resources. A set of topology navigating tools is provided. Alarm situations are indicated by View with a visual indicator such as a change to the object's icon. The topology view includes components from SNMP, CMIP, and Novell NetWare Services Manager managed networks, and may include additional user-supplied data in the future.

The LAN NetView Manage V1.0 View component integrates applications as well as topology. Following an object-action management paradigm, a user selects an icon for management and selects the list of tools to apply to the resource. The View component presents a list of tools that apply to that type of resource. The tool list may include any tools available: from IBM or ISV or home grown, using CMIP, SNMP, or other protocols, etc. The View component transparently invokes the tool (most often an application) by using the command line interface to start the procedure and pass relevant parameters, including the object name of the icon selected by the user. All applications that can be invoked from a command line may be integrated at this GUI level, whether they use any additional Manage facilities or not. Once invoked an application controls its own GUI using SOM objects or roll your own techniques.

Additional services are provided as part of LAN NetView Manage 1.0 to:

- Provide the means to browse SNMP MIBs
- Provide the ability to launch various applications from the topology display such as:
  - Novell NetWare Services Manager 1.5
  - IBM Distributed Console Access Facility* (DCAF)
  - IBM LAN NetView Management Utilities for OS/2 (LMU)

- Provide access to the OS/2 command line on a remote system and return results.

The OS/2 2.x agent, the LAN Requester agent, and Performance data collection agent are all packaged with Manage Version 1.0. and Enabler Version 1.0. A listing of example resources that can be managed by these agents is provided as part of the Enabler description that follows.

2.3.1.3 Customer Value

The major customer benefits derive from the industry standard protocols, API, and from the integration of applications at the user interface level. The standard API allows development of systems management applications which can be ported to other platforms. This encourages a large set of applications to be available from ISVs as well as from IBM. Support of the industry standard management protocols (CMIP, SNMP) allows this single management platform to be used to manage all the heterogeneous resources on the LAN. End-user interface integration promotes a common look and feel across applications and a common means to invoke applications. These reduce operator confusion and training costs. Portability of applications is also enhanced.

The following are the key customer value messages for the platform part of the Manage product:

- Industry standard programming interface endorsed by SystemView, OSF, and X/Open.
- Supports the two popular management protocols - Common Management Information Protocol (CMIP), and Simple Network Management Protocol (SNMP).
- Provides both LAN and WAN management capability from a single platform.
- Provides discovery and topology services as well as other management services to aid in LAN resource management and application development.

These are the key messages for the View part of Manage:

- Increased Productivity
- Reduced Complexity
- Integrated, Heterogeneous Management
- Platform for growth
- Intuitive

2.3.2 LAN NetView Enabler Version 1.0 - Description

2.3.2.1 Overview

LAN NetView Enabler Version 1.0 provides the management framework for agents written for the OS/2 environment. This product provides services and an architected programming interface to agents. The X/Open Management Protocol Application Programming Interface (XMP API) is an industry standard interface endorsed by the SystemView, OSF, and X/Open groups. The Enabler product is a prerequisite for all OS/2 2.x-specific agents including OS/2 itself, LAN Server 3.0, Extended Services 1.0 Database Manager, and Communications Manager/2.

The LAN NetView Enabler product also includes the management agent for the IBM OS/2 operating system environment. The OS/2 agent provides access to

OS/2 resources and data for systems management applications running on the Manage framework. Customer and vendor applications have the same access as IBM applications to the agent data and services.

2.3.2.2 Functional Description

The LAN NetView Enabler product provides management services to the managed OS/2 systems. It provides similar services to those provided by LAN NetView Manage 1.0 on the managing system. It is the base for the Managed System in the OSI management model. The LAN NetView Enabler product provides the XMP API for use by agents written for OS/2 on the managed system. The IBM OS/2 agent, LAN Requester agent, and the LAN NetView Agents Extended product are written to the XMP API. Customers and vendors may also use the XMP API on the LAN NetView Enabler product to create agents.

The LAN NetView Enabler product provides a filtering capability for agent-generated events. Agents generate events to signal applications. Applications register with the Manage product through the XMP API as to which events from which agents from which systems are to be managed by the application. This data is used to set routing and filtering tables in the LAN NetView Enabler product on the managed systems so that only the events that will be used by applications are actually transmitted to the managing system(s).

LAN NetView Enabler OS/2 Agent: In order to manage the resource managers (for example, the operating system itself, Communications Manager, LAN Server, etc.) on a workgroup LAN, the administrator must have access to the data and operational capabilities of each of the software and hardware resources on the LAN. Access to these resources is via a program called an agent. These agents use the interfaces contained in the resource managers to extract data and request changes on behalf of the management applications. The management applications access the agent capabilities using managed object definitions which describe the characteristics of the resource managers and their resources, the actions that can be taken, and the notifications the agent will emit. An Object Catalog is available for each of the resource managers and their resource definitions.

The agents generally support the starting and stopping of the resource manager, modification of configuration parameters, gathering of statistics, and the changing of their operational state. The agents will emit notifications when certain error, potential error, or other significant events occur. The agents also support topology applications.

Three agents: OS 2.x, DOS, and DOS with Microsoft Windows 3.0 or 3.1, have object definitions that are common to a large degree. This commonality in the objects allows management applications to be written once to access all three. The following resources are examples of objects that may be managed by the OS/2 agent in the LAN NetView Enabler or LAN NetView Manage products:

- Physical machine (for example, model ID, serial number, type, owner information, etc).
- Adapters (such as MicroChannel, etc.)
- Floppy and/or Fixed Disk Drive
- Processor and/or Co-Processor
- Printer
- Logical Serial and/or Logical Parallel Port
- Logical Pointing Device

- Logical Volume
- Keyboard
- Display
- Operating System Process
- Operating System Spooler
- Operating System
- Operating System Thread
- Spooled Printer
- Operating System Spooler Queue
- Operating System Spooler Job

In addition the OS/2 agent has a rich set of metrics for collection of performance data. These include instrumentation for CPU, Memory, Files, FAT Cache, HPFS* Cache, Physical Disk, Printer, and Communications Port. An example of the metrics for the CPU are:

- Time the processor was busy
- Number of threads in the processor ready queue
- Number of times a thread was scheduled to use CPU
- Number of interrupts raised
- Time spent servicing interrupts

The agent for the OS/2 LAN Requester is packaged with the OS/2 agent in the Manage and Enabler products. This allows the Operating System client workstations to be managed without added agent packages.

This LAN Requester agent allows management to control and monitor the state of the requester, get and set both initial and runtime configuration attributes, gather statistics about the requesters' operation, and gather requester vital product data.

2.3.2.3 Customer Value

- One major customer value in the Enabler product is the XMP API. This API allows customers to write agents for their applications. The API allows software vendors to provide complementary products, giving customers a greater choice of management products.
- The customer can set filters (from a managing system) using the LAN NetView Enabler product, for events that agents would generate. This prevents unwanted traffic for systems management on the LAN.
- The OS/2 agent enhances the manageability of the OS/2 family of products. The agent makes OS/2 the most manageable workstation operating system available.

2.3.3 LAN NetView Agents for DOS Version 1.0 - Description

2.3.3.1 Overview

The LAN NetView Agents for DOS product provides the management agents for several operating system environments. These include: IBM DOS 5.0 and 6.1, Microsoft DOS 5.0 and 6.0, and Microsoft Windows 3.0 and 3.1 products. The agents provide access to the systems management applications running on the LAN NetView Manage framework. Customer and vendor applications also have access to the agent data for operating systems using the LAN NetView Manage product and the Object Catalogs which describe the attributes and actions that an agent provides.

2.3.3.2 Functional Description

In the IBM LAN NetView Agents for DOS Version 1.0 product there are two operating system agents that support IBM (or Microsoft) DOS 5.0 DOS 6.1 (IBM), and DOS 6.0 (Microsoft), and Microsoft Windows 3.0 and 3.1. The object definitions for these agents are common to a large degree with the OS/2 agent in the LAN NetView Manage and Enabler products, so there is much commonality in the objects. This will allow management applications to be written once to access all three. The following resources are examples of objects that may be managed with the Agents for DOS product:

- Physical machine (for example, model ID, serial number, type, owner information, etc).
- Adapters (such as MicroChannel, etc.)
- Floppy and/or Fixed Disk Drive
- Processor and/or Co-Processor
- Printer
- Logical Serial and/or Logical Parallel Port
- Logical Pointing Device
- Logical Volume
- Keyboard
- Display
- Windows Process (Windows only)
- Windows Spooler (Windows only)
- DOS (DOS only)
- Microsoft Windows (Windows only)
- Spooled Printer (Windows)

2.3.3.3 Customer Value

This set of agents enhances the openness and range of the LAN NetView Family of products. The agents provide detailed data on the status and configuration of the operating systems.

2.3.4 LAN NetView Agents Extended Version 1.0 - Description

2.3.4.1 Overview

The LAN NetView Agents Extended product includes the agents for the OS/2 LAN Server 3.0, Communications Manager/2 1.0, DB2/2 1.0, and Extended Services for OS/2 Database Manager products. The Database Manager and Communications Manager agents can be used to manage servers, as well as client workstations. The LAN Server Agent is for management of the Server only. (The LAN Requester Agent is packaged as part of the LAN NetView Enabler and LAN NetView Manage products.) The agents are accessed by the systems management applications running on the LAN NetView Manage product. The applications use the agents to manage resources in the LAN Server, Communications Manager, or Database Manager. The object definitions for these agents are documented in Object Catalogs which allow vendors and customers to write applications that can access the data in the agents. The LAN NetView Enabler product is a prerequisite for these agents.

2.3.4.2 Functional Description

The LAN NetView Agents Extended Version 1.0 product satisfies the need for Operating System subsystems to have unique agents to manage their resources. This product includes the agents for the IBM OS/2 LAN Server 3.0, Communications Manager/2 1.0, DB2/2 1.0, and Extended Services for OS/2 1.0 (Database Manager) products. It is used to manage servers and gateways, as well as client workstations with the database manager installed.

2.3.4.3 OS/2 LAN Server Agent

The OS/2 LAN Server Agent provides managed objects that:

- Allow most server and requester services to be started, stopped, configured, statistics gathered if the service has any, and their operational state monitored.
- Support topology applications.
- Emit notifications when certain errors occur.

Specific examples of notifications which the agent will provide are:

- Error and Audit log - almost full, full
- Authentication Failure with Domain Controller
- Disk nearing capacity
- Thread limit reached
- UPS power failure
- UPS battery low

2.3.4.4 Database Manager Agent

The LAN NetView Agents Extended product includes support for IBM's Extended Services for OS/2 Database Manager and DB2/2 products. It will allow the management of the database requester, server, and gateway functions. Some examples of the functions provided by the OS/2 Extended Services Database Manager Agent are listed as follows:

- Backup/Restore/Roll-Forward Database- provides the recovery capability for corrupted databases
- Catalog/Uncatalog Database, Remote Database, Node - stores/deletes the databases location information in the database directory
- Emit Notifications - when certain states change or errors occur notifications will be emitted to the managing applications
- Get Database Status - collects the database status such as the last time for backup, number of users connected, the location, etc.
- Get/Update/Reset Database Manager Configuration - retrieves/modifies the Database Manager configuration parameters such as maximum client I/O block size, maximum server I/O block size, size of remote services heap, number of concurrent active databases allowed, maximum number of remote connections to or from this workstation, maximum number of shared segments allowed per database system, etc.
- Get/Update/Reset Database Configuration - retrieves/modifies the database configuration parameters such as size of lock list, size of buffer pool, maximum number of database file opens per application, maximum number of applications allowed, maximum number of remote connections to or from this workstation, etc.
- Support topology applications.

2.3.4.5 Communications Manager Agent

The Communications Manager Agent provides similar management capabilities for the Communications Manager/2 1.0 product. These functions are listed as follows:

- Inventory query, which includes:
  - Finding a product name, version number, CSD level, and other information
  - Querying communications adapter addresses
- Subsystem control and operation monitoring, which includes:
  - Controlling and monitoring the operational state and status of the following:
    - APPN nodes
    - The Communications Manager/2 subsystem
    - Conversations
    - Data link control
    - Logical links
    - Logical units
    - Transaction programs
    - Transmission groups
  - Activating and deactivating Communications Manager/2
  - Activating and deactivating the following:
    - APPN nodes
    - Conversation groups
    - Data link control
    - Logical links
    - LU 6.2 sessions
  - Querying characteristics of the following:
    - APPN nodes
    - Conversations
    - Data link control
    - Logical links
    - LU 6.2 sessions
    - Transaction programs
    - Transmission groups
- Topology support, which includes:
  - Querying connectivity information for the following:
    - APPN nodes
    - Conversations
    - Data link control
    - Logical links
    - LU 6.2 sessions
    - Transmission groups
- Fault monitoring, which includes:
  - Forwarding, as OSI alarms, all alerts from the following components:
    - APPN
    - LAN
    - SDLC

- Forwarding, as OSI alarms, error logs and error messages from the following components:
  - APPN
  - Communications Manager/2 kernel
  - LAN
  - SDLC

2.3.4.6 Customer Value

The agents provide application access to the information provided by the LAN Server, Communications Manager, and the Database Manager for IBM applications as well as vendor or customer applications. These agents continue the policy of making OS/2 the best managed system available. With these products, a customer can realize extensive management capabilities over all of the vital LAN resources whether from an OS/2 system with applications running on Manage, a NetView host (through the Tie application), or from another system using the CMIP-based protocol for management.

2.3.5 LAN NetView Open Industry Standards

A key element of the success of the LAN NetView family of products is the commitment to use of industry standards throughout the framework and applications. Following is a list of some of the industry standards that are implemented by the LAN NetView product:

- 8824 (X.208): OSI - Specification of Abstract Syntax Notation One (ASN.1)
- 8825: OSI - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)
- ISO/IEC 9595 (X.710): Common Management Information - Service Definition
- ISO/IEC 9596-1 (X.711): Common Management Information - Protocol Specification
- ISO/IEC 10165-1 (X.720): Structure of Management Information - Management Information Model
- ISO/IEC 10165-2 (X.721): Structure of Management Information - Definition of Management Information
- ISO/IEC 10165-4 (X.722): Structure of Management Information - Guidelines for the Definition of Managed Objects
- RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets
- RFC 1157: A Simple Network Management Protocol (SNMP)
- ISBN 1-872630-32-4: X/Open Preliminary Specification P170 8/92.

2.4 LAN NetView Applications Product Descriptions

The architected SystemView framework described above with its common application services, APIs, protocol support, and user interface facilities will provide an environment where the application developers can focus their attention exclusively on solving specific systems management problems, and at the same time gain greater interoperability with other systems management applications. The strategy is to provide a complete suite of systems management applications on this framework for managing the OS/2 LAN environment.

The LAN NetView applications listed below are described in more detail in the sections that follow. Each individual product description is followed by a corresponding "Customer Value" description for that particular product.

- LAN NetView Monitor 1.0
- LAN NetView Fix 1.0
- LAN NetView Tie 1.0
- LAN NetView Scan 1.0
- LAN NetView Start 1.1

2.4.1 LAN NetView Monitor 1.0 - Description

2.4.1.1 Overview

IBM LAN NetView Monitor Version 1.0, an application written to the X/Open Management Protocol (XMP) interface of IBM LAN NetView Manage Version 1.0, enables performance management of LAN NetView-managed OS/2 2.x systems. As a system management application, LAN NetView Monitor 1.0 aids in the daily performance management operations of an enterprise through its data collection, graphing, threshold monitoring, and reporting functions. Data collected by LAN NetView Monitor 1.0 is also useful for capacity planning purposes. Specific capabilities offered by the LAN NetView Monitor 1.0 product include the following:

- Provides a common interface across the IBM LAN NetView family of products by using the View interface of LAN NetView Manage 1.0.
- Automated performance management through the use of user-defined policies that specify resources to be collected, collection schedules, thresholds, and data transfer times.
- Collection of OS/2 2.x and IBM LAN Server and Requester 3.0 resource information.
- Threshold alarm notification that results in actions to display a message, log the alarm to the performance database, or exit to a user-specified program.
- Routing of critical threshold alarms to NetView through the LAN NetView Tie 1.0 product.
- Realtime graphical display of performance data, as well as playback from the database.
- SQL database storage of performance data in an IBM Extended Services for OS/2 or DB2/2 1.0 database.
- Report interface to performance database through IBM Query Manager-based menus.
- Command line interface to policy management and log retrieval functions, enabling remote unattended operation of managing systems.
- First Failure Support Technology/2 (FFST/2) error-handling support.
- CID-enabled for remote installation.

2.4.1.2 Functional Description

The following sections provide more detail on the functions and features provided by LAN NetView Monitor 1.0.

- Integration with View component of LAN NetView Manage 1.0

  LAN NetView Monitor 1.0 is integrated with the View component of LAN NetView Manage 1.0, and conforms to SystemView Integration Level 2 and CUA-91 (Workplace Shell* paradigm). This keeps a common look and feel across the IBM LAN NetView family of products, and provides a common point for launching into the function of various LAN NetView applications.

- Policy-based performance management

  Performance management by the LAN NetView Monitor 1.0 product is policy-based, enabling routine performance monitoring to be fully-automated. Groups of systems to be managed are defined as 'management collections' through the View component of the LAN NetView Manage 1.0 product, and each of these collections has a performance policy associated with it. In this policy, the system administrator defines monitoring characteristics that apply to all the systems in the management collection, such as:

  - Resources to monitor (see the description "Resources Supported" below for more detail)
  - Threshold set and clear levels, and exception actions (see "Threshold/Alarm Facility" below for more detail)
  - Monitoring schedule (the hours of the day during which performance data is to be collected)
  - Data collection interval (how often performance data is read during active data collection)
  - Working set period
  - Logging options: whether or not to log performance data, and log file size
  - Schedule for transferring the performance log back to the managing system
  - SQL database storage options: store at "data collection interval," and/or summarize while storing
  - Database maintenance options: whether or not to perform automatic database maintenance, and data retention periods

  Once a policy is started, an instance of the performance agent is started at each system associated with the policy. This agent performs the following actions as specified in the policy:

  - Data is collected according to the resources and schedule specified in the policy
  - If logging is active, the collected data is logged at the managed system.
  - Any defined thresholds are monitored, and when exceeded, an alarm is generated back to the managing system. If the threshold is defined as "critical," and LAN NetView Tie 1.0 has registered to receive it, the alarm will be converted to an SNA alert and routed to NetView.

  A system can be monitored by multiple active policies, which may originate from one or more managing systems. Each policy may specify different resources, schedules, and thresholds to be monitored. Each active policy has its own set of log files associated with it.

  The transfer of performance data back to the managing system is automatically initiated by the managing system at the time specified in the policy, and upon transfer, the data is automatically summarized into the SQL database. (See "SQL Database Storage and Reports" below for more database detail.) An administrator can also manually transfer individual log files back to the managing system at any time.

- Resources Supported

  The performance agent support is packaged in the LAN NetView Enabler 1.0 product and is always installed with that product. Through this support, LAN NetView Monitor 1.0 collects performance information from the following resource groups:

- Critical OS/2 2.x resources: CPU, Physical Disk (both utilization and capacity), RAM, Files, HPFS/FAT Cache, Printer port, Communications port
- OS/2 LAN Server 3.0 and LAN Requester 3.0 resources

In a policy, the user can specify that individual metrics be collected from a resource group, rather than having to collect all metrics within a group. For example, of the following list of CPU metrics, any or all of these metrics can be individually selected for inclusion in the policy:

- tmNotIdle: Time the CPU was busy, including both task and interrupt time
- ctSched: Number of times a thread was scheduled to use the CPU
- ctInt: Number of interrupts
- tmInt: Time spent servicing interrupts
- frCPUBusy: CPU Utilization (tmNotIdle / Time Interval)
- frCPUInt: CPU Interrupt Utilization (tmInt / Time Interval)

- Threshold/Alarm Facility

  Threshold monitoring allows an administrator to "manage by exception." That is, they have the option to ignore managed systems until an event of interest occurs.

  A threshold value and severity can be specified for any metric included in a performance policy. A "threshold clear" value is also specified, which prevents alarms from re-occurring until the clear value has been crossed.

  During performance data collection at each managed system, realtime data values are compared against defined threshold levels. When a threshold is exceeded, one or more of the following actions is taken at the managing system, depending on the policy definition:

  - Display a message in an "Attention Window."
  - Log an alarm in the performance database so that an alarm report can be generated at a later time.
  - Execute a user-specified program at the managing system.

  Threshold alarms (as well as "disk full" or "log full" conditions) can also be registered to be received by LAN NetView Fix 1.0. Actions can then be taken as specified in the LAN NetView Fix 1.0 action table.

  Critical threshold alarms can also be routed to NetView by registering with the LAN NetView Tie 1.0 product to receive the threshold alarms of a specific policy and node. (See the description of IBM LAN NetView Tie Version 1.0 for more information.) Once registered, the critical threshold alarms from that policy and node will flow to LAN NetView Tie 1.0 (as well as back to the managing system) where they will be converted to SNA alerts and routed to NetView on the host.

- Graphing Facility

  The graphing facility is accessed from the LAN NetView Monitor 1.0 folder and is fully integrated into the Workplace Shell. Both realtime and historical graphing (from the database) are supported.

  - A realtime graph can be displayed for any system-level metric being collected from any node defined in any active policy. Up to 15 resources can be displayed on a single graph, and these resources can be from multiple nodes and policies.

  - For historical graphing, any metric from any policy and any node that is stored in the database can be graphed.

The LAN NetView Monitor 1.0 graphing facility also provides the following usability features:

- The user can specify the width, type, and color of each individual resource line on the graph.
- A scaling factor is associated with each line, so that metrics with different value ranges can be displayed on the same graph.
- Graph lines can be temporarily "disabled" from viewing without stopping the flow of data to the graph facility. This is useful for zeroing in on a couple of resources without cluttering the screen with other resources currently being graphed.

• SQL Database Support and Reports

Data Storage

All performance data, policy definitions, and threshold alarms are stored in either an IBM Extended Services or DB2/2 1.0 database.

Storage of performance data is performed as part of the automatic transfer of log files back to the managing system. The user has the option of storing both the raw collected data as well as a summarized version of that data (according to a summary interval specified in the performance policy).

Database Maintenance

Because large amounts of performance data can potentially be collected and stored in the database, old performance data will need to be deleted periodically. To simplify this task, a database maintenance utility is provided which frees the administrator from having to understand SQL and the underlying structure of the database. Additionally, this utility summarizes existing data into daily, weekly, and monthly summary records so that space taken by the underlying detailed data can be freed. Both automatic and manual maintenance functions are provided:

- Automatic Maintenance

  This support is performed during the database storage that occurs during automatic log file transfer. Daily, weekly, and monthly summaries are calculated. "Old" data is deleted from the database according to retention periods defined by the user in the performance policy.

- Manual Maintenance

  This is a Presentation Manager*-based utility for deleting performance and alarm data from the database. The user simply selects the policy, node(s), and date(s) of interest, as well as the type of data to be deleted: detailed (the data as it was collected), summary, daily, weekly, or monthly.

Reports

By accessing the performance data in the database, the administrator can generate resource utilization, trend-analysis, and workload reports, which can be used for performance analysis and capacity planning tasks. A set of pre-defined reports are provided on a per-policy basis through Query Manager menus and queries. These pre-defined reports include:

- Resource reports
  These reports provide information on base operating system resources and LAN Server and Requester resources.
- Application Reports

These reports provide application, file, and thread-level information
- Policy Report
  This report lists the attributes of all defined policies.
- Alarm Report
  This report lists all threshold alarms that are stored in the database.

For more customized reports, the user can access the database directly through SQL. Data can also be exported from the database in Delimited ASCII and Lotus** worksheet formats for use with spreadsheet and graphing programs that support these formats.

- Command Line Interface

  A full command line interface is provided to the LAN NetView Monitor 1.0 policy management and log transfer functions. This enables remote unattended operation of LAN NetView Monitor 1.0 managing systems. The specific commands supported are:

  - Create/Delete Policy
  - Start/Stop Policy
  - Copy Policy
  - List Policies
  - Get Policy Attributes
  - Add/Delete Node to/from Policy
  - Get Performance Data
  - Get Monitor/Scanner Index

- Error Support

  Errors and messages are handled through First Failure Support Technology/2 (FFST/2) support.

  LAN NetView Monitor 1.0 also supports the capability of automatically restarting monitoring activities on either the managed or managing system, if one or the other should go down.

  - Managed system goes down

    When the managed system comes back up, the managing system is notified and then restarts all performance policies currently started at that managed system.

  - Managing system goes down

    When the managing system is restarted, and LAN NetView Monitor 1.0 is restarted, LAN NetView Monitor 1.0 will search its database to see what policies were active at the time the system went down. These policies will be restarted.

- LAN NetView Monitor 1.0 Installation

  There are two parts to the LAN NetView Monitor 1.0 installation:

  - LAN NetView Monitor 1.0 product code installation

    This step installs the LAN NetView Monitor 1.0 code on the managing system, creates a folder for the LAN NetView Monitor 1.0 programs, and registers the LAN NetView Monitor 1.0 policy management interface with the View interface of LAN NetView Manage 1.0.

    Installation is CID-enabled in support of remote unattended install.

  - LAN NetView Monitor 1.0 Database creation and initialization

Prior to creating and initializing the LAN NetView Monitor 1.0 database, either IBM Extended Services for OS/2 Database Manager or DB2/2 1.0 must be installed. LAN NetView Monitor 1.0 then creates its database and initializes it with the tables and indexes required for storing policy, alarm, and performance information.

This portion of installation is not CID-enabled.

- On-line publication and Helps

  The following publications are provided:

  - IBM LAN NetView Monitor Version 1.0 *Getting Started*

    (Printed publication)

  - IBM LAN NetView Monitor Version 1.0 *Administration Guide*

    (Online publication)

  - IBM LAN NetView Monitor Version 1.0 *Help*

2.4.1.3 Customer Value

System Management: Managing Multiple Systems: LAN NetView Monitor 1.0 benefits system administrators by providing the ability to monitor remote system performance through its data collection, threshold monitoring, graphing, and reporting features. Additionally, since all the performance data is stored in a common SQL database, the administrator can use this repository in analysis of performance trends and problems, as well as in load balancing, capacity planning, and network growth management efforts.

Key features benefiting an administrator are:

- Policy-based performance management that is fully automated for routine monitoring activity.

  Once a performance policy is defined and started, performance data is collected and logged under the control of performance agent support at each managed system. At a time specified in the policy, all log files at managed nodes can be transferred to the managing system and processed into the central SQL database without intervention on the part of the administrator. The administrator can then generate reports at regular intervals to obtain a view of how their systems are performing over time. Even routine database maintenance can be handled automatically through the database maintenance utility.

- Detailed performance metrics enabling accurate identification of problem resources.

  Having accurate data helps to reduce problem isolation, analysis, and performance tuning time. As a result, user downtime may also be reduced. Accurate performance analysis may also reduce unnecessary hardware or software expense that may result from trying to solve the wrong problem (for example, buying a faster system when the problem is a shortage of RAM).

- Threshold monitoring.

  The support for thresholds eliminates the need to "closely watch" individual systems for performance problems. Instead, the administrator can opt to "manage by exception," only paying attention to systems when there is a potential problem.

- An SQL interface to performance data.

The SQL interface enables fully-customizable reports for the purposes of resource utilization analysis, trend analysis, and workload studies. The user can also export the data to other spreadsheet and graphing programs for further analysis.

- Remote unattended support.

  Remote unattended support is provided through the following two capabilities:

  - Command line interface, at the managing system, to policy management and log transfer functions.
  - Routing of threshold alarms to NetView through LAN NetView Tie 1.0.

  This enables one administrator to manage several LAN-based managing systems from a single, central location.

End-User Productivity: Improved Worker Productivity: The following aspects of LAN NetView Monitor 1.0 aid the user in the area of productivity:

- The policy management aspect of LAN NetView Monitor 1.0 is integrated with the View component of LAN NetView Manage 1.0, providing a common interface to the management functions of the IBM LAN NetView family of products. This keeps the number of interfaces to be learned to a minimum.

- The graphing interface provides a visual window into the performance of a system, allowing the user to graph metrics from any policy and node on the same graph. This visual presentation can aid in understanding the behavior of individual resources, and can also illustrate how multiple resources or systems are interacting with one another.

  As a Workplace Shell application, the user familiar with OS/2 2.x notebooks will find it easy to change and select graph settings.

  Other usability features include:

  - The capability to specify the width, type, and color of each individual resource line on the graph.

  - A scaling factor also associated with each line, so that metrics with different value ranges can be displayed on the same graph.

  - The ability to temporarily "disable" individual graph lines from displaying without stopping the flow of data to the graph. This is useful for temporarily zeroing in on a couple of resources without cluttering the screen with other resources currently being graphed.

- Installation is CID-enabled for ease of installation of many remote systems.

Investment Protection

Strategic Architectures/Industry Standards: LAN NetView Monitor 1.0 is based on the Open System Interconnection (OSI) managing and managed system model, and is written to the X/Open Management Protocol (XMP) interface of LAN NetView Manage 1.0. Therefore, it is architected to support performance monitoring in a heterogeneous environment. Monitored resources are modeled into GDMO (Guidelines for Definition of Managed Objects) objects, and performance information for these resources are accessed via the object interface of XMP, enabling system-dependent interfaces to performance data to be hidden by the object layer.

Currently, LAN NetView Monitor 1.0 supports only OS/2 2.x systems (at level XR06055 or higher). However, since it is written to standardized interfaces, LAN

NetView Monitor 1.0 is readily extendible to support agents for heterogeneous systems if the agents model their resources into GDMO objects and are written to the XMP interface.

LAN NetView Monitor 1.0 is also integrated with the View interface component of LAN NetView Manage 1.0, and conforms to SystemView Integration Level 2 and CUA-91 (Workplace Shell paradigm). This keeps a common look and feel across the IBM LAN NetView family of products, preventing loss of learning investment.

2.4.2 LAN NetView Fix 1.0 - Description

2.4.2.1 Overview

The IBM LAN NetView Fix Version 1.0 application is designed to receive and process CMIP (Common Management Information Protocol) notifications and SNMP (Simple Network Management Protocol) traps in an OS/2 2.x environment.

The LAN NetView Fix program offers the following functions:

- Register for CMIP and SNMP event notifications from specified resources on selected managed systems
- Store events specified by the user into an IBM Database Manager database
- Display events specified by the user on an Event Console
- Provide special handling for events that are designated as important by the user
- Retransmit events so they can be received by other managing applications (on the same workstation or on a remote workstation)
- Call a pager when a specified event is received
- Display a message pop-up when a specified event is received
- Invoke user-specified routines for personalized handling of received events
- Event Log Browser to allow the user to selectively retrieve and display events from the event log

2.4.2.2 Functional Description

The LAN NetView Fix 1.0 application enables recovery automation for CMIP notifications and SNMP traps. The events to be received and the actions to be taken are controlled by the user. The user specifies the recovery action for the received event in terms of the resource class, event type, time range, and the system from which the notification was emitted. Upon receipt of a notification, LAN NetView Fix 1.0 compares the incoming event with the criteria specified in the action table. If the received event matches an entry in the action table, LAN NetView Fix 1.0 automatically invokes the action(s) associated with that action table entry.

An Application Programming Interface (API) and Command Line Interface are provided, which allows OS/2 applications on a managed workstation to generate and emit CMIP notifications that can be received and processed by LAN NetView Fix 1.0.

All the tasks performed by the LAN NetView Fix 1.0 program occur as the result of an action being invoked. Actions are routines that take event information as input and perform some task (for example, log the event, display the event, beep the pager, etc). Users specify (via the LAN NetView Fix 1.0 Action Table) that an action is to be invoked when an event matching the specified criteria is received. Actions can be one of the following:

- LAN NetView Fix 1.0-supplied
- User-supplied

Installing the LAN NetView Fix program

LAN NetView Fix 1.0 can be installed using the following three approaches:

- Locally from the OS/2 command prompt on the managing workstation using diskettes.
- Remotely on the managing workstation from another workstation on your LAN (via a redirected drive using LAN Server).
- Unattended CID installation.

Starting and Stopping the LAN NetView Fix 1.0 program

LAN NetView Fix 1.0 can be started by using one of the following methods:

- LAN NetView Fix Event processor program icon.
- From the OS/2 command prompt.

LAN NetView Fix 1.0 can be stopped by using one of the following methods:

- From the LAN NetView Fix 1.0 status window.
- From the OS/2 Command Prompt.

Alternatively, the command line interface can be used as an alternate method to invoke some functions such as installing the LAN NetView Fix program on a target workstation, configuring the LAN NetView Fix program, starting and stopping the LAN NetView Fix program, enabling/disabling trace, and invoking the CMIP Notification Emitter.

LAN NetView Fix 1.0 has 2 kinds of information available on-line:

- The online LAN NetView Fix Administration Guide
- Message helps

A printed LAN NetView Fix 1.0 "Getting Started" guide provides installation and start-up instructions.

2.4.2.3 Customer Value

User Productivity - Recovery Automation: LAN NetView Fix 1.0's chief benefit to the end-user is that it enables recovery automation for CMIP notifications and SNMP traps. The events received and actions taken are controlled by the user. The user specifies the recovery action for the received event. Upon receipt of a notification, LAN NetView Fix 1.0 looks up the action table and automatically invokes the action(s) associated with the matching action table entry. LAN NetView Fix 1.0's coordination of fault detection and recovery will minimize both costly downtime and the resources required to identify and correct problems, and help to avoid the cost of taking erroneous actions.

User Productivity - Improved Worker Productivity: The LAN NetView Fix 1.0 product enables a system administrator to manage network errors and notifications (CMIP and SNMP) from a single managing workstation and keep a central log of selected errors and notifications generated on the network.

System Management: Open Architected Platform: The LAN NetView Manage 1.0 platform's use of industry, international, and IBM standards and architectures allow customers to support a wide array of devices. LAN NetView Fix 1.0 itself is open and architected so customers can easily extend the product to implement management operations and recovery procedures. This flexibility enables a customer to tailor LAN NetView Fix 1.0 functions to a particular environment, as well as to extend or enhance LAN NetView Fix 1.0.

Investment Protection - Open & Architected Platform: LAN NetView Fix 1.0's use of LAN NetView Manage 1.0 platform services as well as the extensibility of the LAN NetView Fix 1.0 application functions, will protect both end-user service levels and the customer's system management investment.

2.4.3 LAN NetView Tie 1.0 - Description

2.4.3.1 Overview

The IBM LAN NetView Tie Version 1.0 product allows a NetView operator to receive notifications from the network resources managed by the IBM LAN NetView family of products. LAN NetView Tie 1.0 transforms OSI alarm notifications to SNA alerts. Non-alarm notifications are wrapped in an Event Major Vector. Both types of notifications are sent to the host system via IBM Extended Services for OS/2 Communications Manager or Communications Manager/2 1.0.

2.4.3.2 Functional Description

LAN NetView Tie 1.0 provides both OS/2 and NetView command line interfaces. LAN NetView Tie 1.0 receives commands entered from the NetView command line V1.2 RUNCMDS, using the Remote Operations (ROP) Services. Following are the types of commands supported by LAN NetView Tie 1.0:

- Start and Stop commands

  Start and stop commands are supported. LAN NetView Tie 1.0 can also be started from OS/2 by using the LAN NetView Tie program icon. Two types of stop commands are supported, for both graceful and immediate termination of activities.

- Register and Deregister commands

  These commands allow the user to selectively specify which notifications LAN NetView Tie 1.0 is to receive from LAN NetView-managed resources. The REGISTER command provides the capability of selecting notifications based on node, object class, event type, object instance, time, and number received in a specified timeframe. The result of a REGISTER command is to set up an "event sieve." The DEREGISTER command deletes event sieves. These commands allow a NetView operator to filter the notifications to be received from a particular resource.

A configuration file (flat ASCII file) can be created that contains a base set of event sieves. This file can be used to cause LAN NetView Tie 1.0 to automatically register for the notifications at program startup. During LAN NetView Tie 1.0 execution, a runtime configuration file is created that is initialized from the base set of event sieves and is updated to reflect the changes resulting from REGISTER and DEREGISTER commands. Either configuration file can be selected on a restart of the LAN NetView Tie program after it has been stopped.

LAN NetView Tie 1.0 can register to receive both alarm and non-alarm notifications. LAN NetView Tie 1.0 converts OSI alarm notifications to SNA alerts or resolutions. A file is provided that contains a set of object identifier (OI)-to-code point mappings. This file is used to map standardized and IBM-architected OIs to SNA/MS alert code points. LAN NetView Tie 1.0 supports mapping the following five OSI alarm types defined in ISO standard 10164-4 to SNA alerts or resolutions:

- Communication
- Quality of Service
- Processing Error
- Equipment
- Environmental

LAN NetView Tie 1.0 also supports the creation of user-defined OI-to-code point mappings in ASCII files. An OI-to-Codepoint Compiler, shipped with LAN NetView Tie 1.0, is used to create a binary OI-to-codepoint mapping table from the ASCII file. Both the IBM-provided mapping table and optional user-defined tables are used for mapping the alarm object identifiers to alert code points.

Non-alarm Common Management Information Protocol (CMIP) events are wrapped in an Event Major Vector which contains the parsed CMIP event type and the managed object class along with the event information encoded according to Basic Encoding Rules (BER). The Event Major Vector is sent to a host system where a user-supplied application could parse the BER-encoded event information.

LAN NetView Tie 1.0 sends the alerts and/or resolutions to NetView where they can be converted and displayed on the screen in readable text. The IBM Communications Manager SNA/MS transport services are used to send the alerts, resolutions, and events to the host system.

The following problem determination aids have been implemented in LAN NetView Tie 1.0 to support the resolution of problems in a timely manner:

- Messages - Informational and error
- Command Line Message Helps - Provide causes and recommended actions
- Probes - Utilize the message log, error log, and dump functions of the FFST/2 program to report errors and collect error data.

LAN NetView Tie 1.0 can be installed in three ways:

- Locally from an OS/2 command prompt on the managing workstation.
- Remotely to the managing workstation from another workstation (via a redirected drive using LAN Server).
- Unattended CID installation.

2.4.3.3 Customer Value

IBM LAN NetView Tie Version 1.0 improves centralized control of LAN environments by providing a means for sending CMIP notifications to NetView in a manner that NetView can understand. The NetView operator can register event sieves (or filters) to receive specific notifications from LAN resources. These filters can be changed via REGISTER or DEREGISTER commands from the NetView command line interface or local OS/2 command line interface. The NetView operator can register to receive both alarm and non-alarm notifications. LAN NetView Tie 1.0 reports the problems and clearing of problems identified in the alarms by converting the OSI alarms to SNA alerts and sends them to NetView via IBM Extended Services Communications Manager, or IBM OS/2 Communications Manager 1.0. The alarm information can then be displayed at a NetView console in a readable form. The fault information is provided in a manner that could be used by automation routines running on NetView. LAN

NetView Tie 1.0 also wraps non-alarm notifications in a new Event Major Vector and sends them to a host system where a user-supplied application can parse the information. This means that LAN NetView Tie 1.0 allows NetView running on a host system to provide better centralized control for different LAN environments.

2.4.4 IBM LAN NetView Scan - Description

LAN NetView Scan is currently available through a beta program. The announcement of this product's availability will be determined by the experience of the Beta participants and the feedback they provide to IBM on the LAN NetView Scan product's function and usability.

Like the LAN NetView Monitor, LAN NetView Tie and LAN NetView Fix applications, the LAN NetView Scan product is designed as a tool for systems management. Specifically, the LAN NetView Scan product provides function for configuration and inventory management of LAN-attached workstations running OS/2, DOS and DOS with Windows.

Here are some of the features of the LAN NetView Scan product:

- The LAN NetView Scan product collects workstation hardware inventory data (sometimes referred to as Vital Product Data or VPD) and stores it in a centrally-located SQL database. The convenience of a centralized inventory database can be exploited for report generation using database query tools or custom applications, and for planning hardware and software upgrades when existing workstation resources need to be considered.
- The LAN NetView Scan product monitors the status of workstation files. The specific files to monitor are selected by the administrator. It monitors the files for changes and when change is detected, collects the files and/or information about the files (filesize, date/time stamp) into a SQL database called the "Filestore". The LAN NetView Scan product's filestore can hold multiple versions of the file data.

  This feature of the LAN NetView Scan product can be used to manage critical workstation configuration files such as CONFIG.SYS, or AUTOEXEC.BAT, or the output of other workstation utilities such as backup or diagnostic programs. Similarly, the LAN NetView Scan product can be used to monitor the file statistics of key application executables for version tracking.

  Monitoring files at OS/2 workstations can be scheduled to take place at designated times. Monitoring files at DOS or DOS with Windows workstations cannot be scheduled, but can be automated through convenient initialization procedures such as AUTOEXEC.BAT or a login profile.
- The LAN NetView Scan product provides a command scheduler. At regularly scheduled times, it will run programs or commands at selected OS/2 workstations to perform such tasks as software inventory, system backup, virus checking, system diagnostics and report generation.
- The LAN NetView Scan product provides an application exit facility. For customized, post-processing of any data that it collects, the LAN NetView Scan product supports application exits written as dynamic link routines (DLL), command files, REXX routines, or executables (EXE).

2.4.5 LAN NetView Start Version 1.1 - Description

2.4.5.1 Overview

The IBM LAN NetView Start 1.1 program is a tool for planning and managing the configuration of OS/2 software and for enabling IBM configuration, installation, and distribution (CID) conventions for remote installation in an OS/2 LAN.

The LAN NetView Start program provides an object-oriented, graphical, Presentation Manager interface enabling an administrator to plan and manage the configuration of network workstations within the intuitive context of a graphical representation of the network. For each workstation requiring a change in configuration (selected by the administrator), the LAN NetView Start program generates the files required by the IBM CID process that enable remote (across-the-network) installation or automated program distribution. These files include configuration response files and either OS/2 REXX install command files for use with NTS/2, or IBM NetView Distribution Manager/2 (NvDM/2) change files depending on the method of program distribution.

The following IBM subsystems are supported by the LAN NetView Start program in each type of CID file:

- LAN Server 3.0, Entry and Advanced
- Extended Services for OS/2
- Network Transport Services/2

In building the NetView DM/2 change files for automated code distribution, the LAN NetView Start program can include both CID-enabled applications as well as those applications not enabled for the CID process. Only applications enabled for CID are included in the REXX install command files, which are used by the IBM Network Transport Services/2 product for remote installation.

The most significant enhancement in Version 1.1 of the LAN NetView Start product is an administrative interface to the IBM NetView Distribution Manager/2 product to generate and catalog NetView DM/2 change files. During the process of building the change files, the LAN NetView Start program also creates two cross-reference lists, mapping code server and change files to workstations to further assist the administrator in managing workstation updates.

Version 1.1 of the LAN NetView Start program also includes a number of changes designed to enhance its usability. These include:

- An applications union window that enables the administrator to identify and manage the application content of groups of workstations simultaneously.
- A node list notebook that allows the administrator to manage supplemental response files for applications across selected groups of workstations.
- Object pop-up menus similar to those provided in the OS/2 2.x desktop.

Finally, Version 1.1 includes three mini-applications for preparing the Network Transport Services/2 code server, for converting plain-text (ASCII) database files into SQL database rows, and for generating a master attribute value file from a SQL database. (Note: IBM is providing these utilities as a convenience to the user. They are provided "as is" and are not included in the service described for the program product.)

With the availability of Version 1.1 of the IBM LAN NetView Start product, Version 1.0 was withdrawn, effective immediately. The IBM LAN NetView Start product was initially released as LANfocus Start/2.

(Note: LAN NetView Start Version 1.1 does not use LAN NetView Manage 1.0)

2.4.5.2 Functional Description

The LAN NetView Start program provides an object-oriented, Presentation Manager, graphical user interface for managing the configuration of OS/2 software in a LAN environment. Using the LAN NetView Start program, networks are composed of subnetworks or topologies. A topology can represent any logical segment of the network such as a department or a floor, or it can represent the network in its entirety. The software configuration of workstations is managed within the context of the network topology.

Creating a Network Topology: The LAN NetView Start program provides two methods of creating topologies: using the drag/drop technology of the user interface or migrating existing configuration information.

With the user interface, a network topology is created by dragging and dropping a workstation object into a topology drawing area, defining the workstation's software and functional (and some hardware) characteristics, and drawing the desired connections to other nodes.

The LAN NetView Start program is capable of representing workstations with the following characteristics:

- Operating System
  - OS/2 Version 2
  - OS/2 Extended Edition* Version 1.3
  - OS/2 Standard Edition Version 1.3
  - DOS
  - DOS with Windows
- LAN Function
  - Domain Controller
  - Additional Server
  - OS/2 Requester
  - DOS LAN Requester
- Database Function
  - OS/2 Database Server
  - OS/2 Database Client
  - OS/2 Database Server and Client
  - DOS Database Client
- 3270 Emulation (up to 10 sessions; 5 DFT, 5 non-DFT)
- Off-LAN (Host) - Token Ring, SDLC, DFT
- Applications or maintenance updates
- Adapter types
  - 3270 - DFT, SDLC
  - LAN - Token Ring, Ethernet**

During the creation of the topology, the LAN NetView Start program interactively validates connections drawn between remote data services servers and clients, and between LAN Server servers and requesters ("If you can draw it, it's valid.").

The LAN NetView Start program stores the resulting network configuration information either in a plain-text (ASCII) file or in a SQL database, which requires the Database Manager component of the Extended Services product.

The LAN NetView Start program also includes a capability for migrating existing workstation configuration data into the LAN NetView Start topology database.

The node data collector utility is used to collect configuration data at workstations currently running:

- OS/2 Extended Edition 1.3
- OS/2 Standard Edition 1.3 with Extended Services or LAN Server 2.0
- OS/2 2.0 with Extended Services or LAN Server 2.0

The node data collector utility builds an import file for each workstation. The import file can be created on the hardfile of the workstation, on a diskette, or the creation can be redirected to a server. When the files are made available at the workstation, the administrator can use the graphical facilities of the OS/2 2.x Workplace Shell to display them as a folder of individual file objects. To integrate the import file data into the LAN NetView Start program's topology database, the administrator simply drags an import file object, a group of objects, or the folder object and drops it on the topology.

In this way, existing networks can be represented graphically by the LAN NetView Start program.

IBM Configuration Installation Distribution (CID): In addition to providing an interface for managing network configuration information, the LAN NetView Start program enables the implementation of IBM CID conventions for automated program distribution, in conjunction with the IBM NetView Distribution Manager/2 product, or remote software installation via the IBM Network Transport Services/2 product. To support the CID process, the LAN NetView Start program generates configuration response files and either REXX install command files for use with the Network Transport Services/2 product, or change files for use by the NetView DM/2 product.

The following IBM subsystems are supported in each type of CID output file:

- LAN Server 3.0-Entry and Advanced
- Extended Services for OS/2
- Extended Services with Database Server for OS/2
- LAN Adapter and Protocol Services component of the Network Transport Services/2 product

In building the NetView DM/2 change files for automated code distribution, the LAN NetView Start program can include both CID-enabled applications as well as those applications not enabled for the CID process. Only applications enabled for CID are included in the REXX install command files, which are used by the IBM Network Transport Services/2 product for remote installation.

The REXX files generated by the LAN NetView Start program contain commands to invoke the install programs of the software targeted for the workstation and are designed to be processed by the LAN CID utility of the Network Transport Services/2 product. One REXX file is built for each workstation requiring an installation or configuration update and is built when the administrator chooses; for a single workstation or group of workstations, for a single topology or group of topologies.

In the generation of REXX files, the LAN NetView Start program supports CID-enabled applications in addition to the OS/2 subsystems. One of the objects in the LAN NetView Start program's primary folder is the applications folder, provided as a convenient means for keeping track of the applications. Multiple application folders can be created or all the application definitions can be maintained in one folder.

A response file is a plain-text file containing keywords and values that direct the installation and/or configuration of software without requiring the presence of the workstation user to respond to install prompts. For each workstation requiring an installation or configuration change to an OS/2 subsystem, the LAN NetView Start program uses a set of tuning algorithms to calculate values for key configuration parameters and build a response file for each subsystem. In calculating the parameter values, the LAN NetView Start program considers:

- Function (server, requester, etc.)
- The requirements of co-resident software (including applications)
- The requirements of dependent workstations (for example, the number of LAN Requesters with connections to the Server)
- Adapter type

The LAN NetView Start program does not generate values for all available product parameters; only those that need calculating based on the total software composition of the workstation and its position in the network. To affect changes to the parameters not generated by the LAN NetView Start program, the administrator can imbed supplemental (user-defined) response files within those generated by the program. Furthermore, the administrator can specify whether the supplemental file should be appended at the end or inserted at the beginning. Appending a supplemental response file at the end allows overriding any values calculated by the LAN NetView Start program.

As with the REXX files, the response files are generated when and as the administrator chooses; for a single workstation or group of workstations, for a single topology or group of topologies.

2.4.5.3 Customer Value

Usability: The value of the LAN NetView Start program can be expressed in terms of its ability to manage network configuration information as well as its contribution to the CID environment. The user interface, compliant with SystemView Integration Level 2, includes many features designed to enhance the products usability as a system management tool above and beyond the inherent usability of an object-oriented, graphical user interface.

Some of the usability features include:

- Notebook controls

  The LAN NetView Start program organizes the storage of network configuration information by notebook. Each network, topology, node and application, as well as the primary LAN NetView Start 1.1 folder, the delete folder, the transformers and any user-created folders have a separate notebook. In most cases, the notebook of a particular object is presented when the user double-clicks on the object icon. Every notebook has the same controls for paging through the information stored within it.

- The applications union window

  The applications union window is a convenient way of managing the application content of groups of workstations simultaneously. Within the applications union window, after the administrator has selected multiple workstations, a single icon is displayed for any application appearing on at least one of the selected workstations. At this point, applications can be edited, added to other workstations in the group, or deleted from the entire group. New applications can be introduced into the group by dropping an application object on the applications union window.

- Object pop-up menus

  As with the OS/2 desktop, the LAN NetView Start program supports a pop-up menu for each LAN NetView Start 1.1 object. The pop-up menu is invoked by positioning the pointing device over the desired object and clicking the menu button.

- View-oriented presentation

  For simplicity and manageability, the LAN NetView Start program organizes the presentation of the network configuration into three connection views: the LAN view, the Database view and the 3270 view. Each view presents information relevant only to that view. In the LAN view, for example, workstations with no LAN function are displayed as generic node icons overlayed by the international "not" symbol (circle with a slash). LAN workstations are displayed with the appropriate LAN icon and labelled with the node name or user name or other detail as chosen by the administrator. The presentation while in the other views is similarly selective.

- Customizable node templates

  To facilitate the creation of a network topology, the administrator is allowed to create customized workstation models or templates, which serve to minimize the amount of customization required when the workstation object is dropped into the topology drawing area.

- Node duplication (including connections)

  If the configuration of a particular workstation, including its connections, is significantly representative of other workstations, the administrator can choose to replicate it up to 99 times. The LAN NetView Start program will automatically generate the necessarily unique attributes of each duplicate (like node name, etc.)

- Find a Node

  The find a node function is useful when your topology is large enough that you have difficulty finding a specific node. The procedure works like a text search in a word processor and can search a variety of fields in the node settings notebook. After entering up to 60 characters for a search string, the administrator chooses a radio button indicating what field of the node notebooks to search. The radio buttons include:

  - Search node names
  - Search node comments
  - Search LAN adapter addresses

- Search user names
- Search machine locations

- Autodraw

  With autodraw enabled, connections are automatically drawn from workstations newly dropped into the topology to the most recently indicated server. ,*

- Deletions

  The act of deleting an object from the network is simplified by allowing the object to be dropped on the delete folder. Until the LAN NetView Start program is stopped (or the machine is powered down), the information on any deleted item is kept as an object in the delete folder. To reinstate the object, the user simply drags it from the delete folder and drops it back in the topology.

- Transformers

  The LAN NetView Start program supports the use of transformer objects to simplify the act of requesting CID output. Separate transformer objects are provided to generate response files only, both response files and REXX install command files or both response files and NetView DM/2 change files.

  Using the transformers, CID output can be requested for a specific workstation, a group of workstations, a specific topology or for a group of topologies simply by dragging the object(s) (the workstation icon, for example) and dropping it on the appropriate transformer.

- Transformer status lists

  Each transformer provides a status view that displays four status lists used to manage and track transformer progress per workstation. The four status lists are:

  - Queue - lists the workstations yet to be processed (in the order in which they will be processed).
  - Success - lists the workstations successfully processed.
  - Warning - lists the workstations for which output was successfully built in spite of some anomaly.
  - Reject - lists the workstations for which output was not successfully built.

  The status view includes other facilities for resolving the causes of warnings and rejections.

  It is through the status view that the administrator is able to get to the output files for viewing or editing.

Installability: The installation of the LAN NetView Start program is a one-step process and manages the necessary changes to the system files as a part of the installation process.

The LAN NetView Start program is also CID-enabled.

Maintainability: To provide for convenient problem determination and problem source identification, the LAN NetView Start program generates error logs using first failure strategy. Each log entry identifies the component and module that encountered the error and, where appropriate, the service that was requested and the condition code the service returned.

Adaptability: Adaptability is enhanced through the support of applications.

Additionally, the LAN NetView Start program allows the use of a user-supplied attribute value file, which enables an administrator to control workstation attributes consistent with existing conventions. The attribute file is a plain-text file with attribute names and corresponding values in a format defined by the LAN NetView Start program. To implement the values contained in the attribute file, the administrator drops the attribute file object on the desired topology.

The attribute file can be used to control such attributes as:

- Token Ring address
- SNA node ID
- Local node name and alias
- Database workstation name
- LAN Computername

Exploitation: The LAN NetView Start program enables the exploitation of the system management features, for example, CID-enablement, of the OS/2 subsystems and applications.

Also, the LAN NetView Start program exploits the usability features of the OS/2 2.x Workplace Shell providing additional consistency across user interfaces, which contributes to the overall manageability of the system.

Standards and Architectures: The LAN NetView Start program's user interface is compliant with the SystemView Integration Level 2.

Investment Protection: The LAN NetView Start program is designed to enhance the manageability of OS/2 2.x software in a LAN environment without requiring additional skill levels or staffing.

Functionality: For those customers with a current investment in OS/2 2.x, the LAN NetView Start program is designed to reduce the workload as well as the knowledge-level required of the system administrator and support staff for configuration management, which should enable the customer to manage existing environments with a smaller support staff or grow with the current staffing.

For customers with investments in earlier versions of OS/2, who intend to upgrade, the capability of the LAN NetView Start program to migrate existing data into the LAN NetView Start 1.1 database enables the upgrade with existing skills and hardware.

Adaptability: The adaptability of the LAN NetView Start program, as described in the System Management Adaptability section (support for applications; supporting a user-supplied attribute value file), should enable customers to manage existing network resources with existing skills and hardware.

Standards and Architectures: The LAN NetView Start program's user interface is compliant with SystemView Integration Level 2 and exploits the usability features of the OS/2 2.x Workplace Shell, which provides for user interface consistency across products minimizing the necessity of additional education or training.

End-User Productivity: In general, the LAN NetView Start program is designed to enhance user productivity by reducing the amount of time and effort spent on configuration management. The LAN NetView Start program is designed to enable both the network administrator and the workstation user to realize increased productivity.

To maximize the benefit to personal productivity, it is highly recommended that the LAN NetView Start program be used on a high-end 486-based personal computer.

Functionality: The LAN NetView Start program provides for centralized system configuration management, displacing the need for the administrator spending time at individual workstations gathering or manipulating configuration details.

The interactive validation of connections drawn between remote data services servers and clients, and between LAN Server servers and requesters, and the ability of the LAN NetView Start program to calculate configuration parameter values for the individual products using knowledge of the complete configuration, together serve to reduce or eliminate the need for the "trial and error" approach to workstation configuration. As a result, the end-user should realize increased productivity through a reduction in downtime associated with configuration or installation changes.

Usability: As a result of the extensive usability features of the LAN NetView Start program, as described in the System Management Usability section, the administrator should realize increased productivity through the reduction of time spent on configuration management.

Installability: The LAN NetView Start program provides support, to the extent possible as a CID tool, for both CID-enabled applications and applications not enabled for the CID process. This support enhances the installability of applications, which should further contribute to user productivity.

Exploitation: Through compliance with SystemView Integration Level 2 and exploitation of the usability features of the Workplace Shell introduced with Version 2 of OS/2 operating system, the LAN NetView Start program provides a common, consistent user interface, which should ease the adoption of the program without impacting productivity.

2.5 LAN NetView Applications from Other Vendors

2.5.1 NetWare Services Manager

Novell Inc. has shipped NetWare Services Manager 1.5 for OS/2 (for the LAN NetView product), an IBM OS/2-based application that lets users manage PCs running Novell's NetWare operating system from a workstation where LAN NetView Manage 1.0 is running. In conjunction with LAN NetView Manage 1.0, NetWare Services Manager 1.5 lets network administrators manage NetWare nodes and IBM LAN Server nodes from a single management console. This enables the LAN NetView family of products to address the requirements of Netware 3.11 and 4.0 users. The IBM LAN NetView user interface will be used to view the LAN topology, providing a physical view of the LAN which includes the NetWare servers and requesters. The user would then be able to select the NetWare server and access the NetWare Services Manager 1.5 functions.

2.5.2 LANlord

Microcom Inc. is developing a version of their LANlord desktop management system for the LAN NetView platform. This version of the LANlord product will provide users with access to a fully-integrated set of applications for managing DOS, Windows, and OS/2 LAN-based PCs from the LAN NetView View component, and will utilize the LAN NetView Manage platform services. The LANlord product on the LAN NetView platform will provide integrated applications for centralized, remote desktop management, including realtime PC hardware and software discovery and inventory, client and server monitoring, remote configuration, software metering, and reporting of statistics to facilitate management of network nodes.

2.5.3 Network Analysis Series

ProTools Inc. is integrating their LAN performance analysis application with the LAN NetView framework. Network Analysis Series is an integrated solution set for monitoring, characterization, and analysis of distributed networks. Two applications, Foundation Manage**r and Cornerstone Agent**, comprise the Network Analysis Series. The Foundation Manager product is not only a full-function network management system in its own right, but also a central console for viewing and controlling subnets throughout an enterprise. The Cornerstone Agent product, which is a SNMP RMON (Remote Monitoring MIB) agent, executes realtime monitoring and analysis functions under control of a local administrator or remote console like the Foundation Manager product. The two products work together to form an enterprise-wide network management system.

Monitoring and filtering network activity, analyzing protocols, setting up alarms, and displaying statistics can be executed locally or remotely. This enables a network administrator to monitor the health of any network and pinpoint problems before they occur. This is accomplished through protocol analysis, which is basically the function of decoding protocols such as TCP/IP, NetBIOS, etc., from cryptic notation into a readable representation appropriate for reporting statistics, and/or passing to graphics applications to produce charts and graphs.

The net effect of this integration is that users of LAN NetView 1.0 will have the ability to monitor and analyze local or remote networks from a single platform. The Foundation Manager product will be able to be invoked from the LAN NetView Manage 1.0 user interface, and will also share data from the LAN NetView 1.0 Topology/Discovery service.

2.5.4 AlertVIEW

Shany Inc. plans to deliver a version of their AlertVIEW product for the LAN NetView framework. The AlertVIEW program acts as a network "sniffer" for applications; using agents to look inside the software to detect and analyze errors, or potential error conditions. Error messages can be monitored by the network manager, and along with hardware and software configuration data that is forwarded by the AlertVIEW program, allows corrective action to be taken immediately, sometimes even before users are aware of the error conditions. The AlertVIEW program also allows a network manager to launch an application on a remote workstation which will run in the background to correct a problem. Actions like this can also be automated. Other capabilities provided by the

AlertVIEW product include: filtering of events, automatic discovery of new agents, virus detection, asset management, and an SNA-Gateway function.

2.6 Configuring LAN NetView Workstations

In preparing to install LAN NetView products on your network of workstations, some planning steps are required. Assuming that the initial decisions of what resources are to be managed and what LAN NetView applications are required to satisfy these management requirements have been made, the decision must then be made as to which workstation(s) are to be designated as "managing" systems and which workstations are to be designated as "managed" systems.

The "managing" system(s) can be viewed as systems management servers for the "managed" systems. The decision as to whether more than one managing system or how many managing systems are required will be governed by several factors, as with other types of servers. For example, the number of workstations involved, the nature and extent of management activity planned, the physical locations of the LAN workstations (dept., site-wide) are some of the consideration factors. Keep in mind the managing systems must be OS/2 workstations.

Once the designation of "managing" and "managed" nodes is completed, installation of the appropriate LAN NetView products and components for each workstation can begin. All of the LAN NetView family of products are CID-enabled to make the installation process easier, quicker, and more convenient.

2.6.1 Managing OS/2 Workstation

Each managing system has, or will require, the OS/2 2.x operating system software installed as the software base for LAN NetView 1.0.

The first LAN NetView product to be installed will be LAN NetView Manage 1.0, which provides the infrastructure and common services required for all LAN NetView management applications. It also contains the OS/2 resource agents to allow the operating system resources of the managing system to be managed along with other managed systems on the LAN (that is, the managing system can manage itself). Next, install the View component, the user interface. If the managing system also contains any of the OS/2 system extension software (LAN Server, Communications Manager/2, Database Manager or DB2/2) then the appropriate agent components of LAN NetView Agents Extended will need to be installed to allow those system resources to be managed. LAN NetView Manage 1.0 also contains the OS/2 LAN Requester agent as part of the offering.

Next the selected LAN NetView applications can be installed. The sequence of installation will not matter in most cases. If applications have established prerequisite relationships, this will be spelled out in the installation instructions of those specific applications. Install all applications selected for this managing station at this time, whether supplied by IBM, a vendor, or internally developed.

When all of the LAN NetView applications are installed on the managing system, it is time to start managing. If other managing systems are required, they can be installed next, otherwise the managed systems can be installed. Other managing systems may either have the same set of LAN NetView applications installed or may be configured quite differently.

2.6.2 Managed OS/2 Workstation

On OS/2 managed workstations, the first LAN NetView product to be installed would be LAN NetView Enabler product. LAN NetView Enabler 1.0 provides the set of common services that support the resource agents. The agents for the OS/2 operating system resources and the OS/2 LAN Requester are supplied as part of the LAN NetView Enabler product. Other sources for OS/2 resource agents include the LAN NetView Agents Extended product or other products that vendors may provide. In instances where customers have developed their own software whose resources can profit by LAN NetView management, agents may be written to exploit the LAN NetView Enabler product's services as well.

Once installation of LAN NetView Enabler 1.0 is complete, the managed OS/2 workstation is functional since it contains the OS/2 agents. If the software installed on the OS/2 workstation also includes the IBM Communications Manager/2, the Database Manager from the IBM Extended Services for OS/2, the IBM DB2/2, or the IBM LAN Server products, the next step should be to install the LAN NetView Agents Extended product. It contains the resource agents being supplied by IBM for each of the aforementioned products.

That completes the installation of IBM supplied LAN NetView products for the OS/2 managed workstation. If other OS/2 resource agents exist, they should now be installed to complete the functionality of the workstation. The workstation is now ready to be managed.

2.6.3 Managed DOS Workstation

For the DOS workstation, the first and only IBM supplied LAN NetView product to be installed is LAN NetView Agents for DOS. It contains the resource agents for DOS 5.0/6.0/6.1 including those for Microsoft Windows 3.0 and 3.1. Unless other DOS agents have been developed or purchased, this completes the LAN NetView installation process for the DOS managed workstation, and it is ready to be managed.

2.7 Positioning of LAN NetView with Other Products

IBM's leadership role in network management can clearly be seen through the success of the NetView and NetView/6000 network management systems. LAN NetView 1.0 builds on the success of these management platforms, providing distributed systems management on the personal workstation. Now, in addition to providing centralized and distributed management of network components from host and RISC-based systems, LAN administrators will be able to manage personal workstations and other LAN resources with a lower-cost LAN NetView product.

When we speak of positioning the LAN NetView family of products with other IBM products we need to emphasize that we're not talking strictly about network management, but rather systems management; which of course is what IBM SystemView is all about. Next, we've got to differentiate between management platforms and applications. NetView on the mainframe, NetView/6000, and LAN NetView are considered platforms, though of course they have applications as part of their product families. LAN Network Manager* and LAN Management Utilities/2 are applications that run on OS/2 without using any management platform services, currently.

One of the questions most often asked by customers, ISVs, and our IBM marketing people in regard to our systems management products is "Which of these products should I use, or recommend to my customer?". The answer depends on many factors: the size of the network, the installed hardware (mainframe, RISC, or Intel**-based), the network node configuration (for example, SNMP devices, types of workstations, etc.), preferred method of management (centralized vs. distributed), importance of managing workstations and network connections, and of course--cost. This is not an all-inclusive list by any means, but as you can see, there are many things to consider. You can also surmise that the customer's requirements may be met by any, or all of the three product families, again depending upon the aforementioned factors.

In general, NetView is used in large enterprises for either centralized network management, or in conjunction with the AIX* NetView* Service Point and/or the NetView/PC* products for distributed management. It is used where there exists a large concentration of SNA devices, and functions as a consolidation point as a Manager of Managers with other networking products. LAN NetView 1.0 interoperates with NetView through the LAN NetView Tie product which allows a NetView operator to monitor and control LAN NetView-attached workstations. Refer to the detailed description of LAN NetView Tie in this paper for further information.

The NetView/6000 product initially focuses on SNMP device management, using applications for fault, configuration and performance functions. It also performs management of RISC workstations. The XMP API is provided for developing systems management applications; the same API used by LAN NetView. Thus, over time, it is anticipated that the NetView/6000 product will be providing additional applications for the management of personal workstations.

LAN NetView 1.0 initially focuses on systems management of OS/2, DOS, DOS with Microsoft Windows, and Novell environments, and enabling multi-protocol support. SNMP device management is provided via the LAN NetView Request Manager component of LAN NetView Manage 1.0. This applet allows users to issue commands to SNMP devices to query the status of the device(s) or to SET values to control the device(s). MIBs are included for RMON (Remote MONitoring) and the IBM 6611 Multiprotocol Router, which can be managed using the Request Manager component of LAN NetView Manage 1.0. However, all other SNMP devices can only be monitored/controlled at the MIB II level, since private extensions cannot be easily compiled into the LAN NetView product's metadata database (LAN NetView MIBs are in GDMO format) with the initial release of the LAN NetView product. It is anticipated that several of the more popular vendors' MIBs will be converted to GDMO format, tested, and distributed for use with LAN NetView 1.0, subsequent to its general availability. Electronic distribution via BBS will be the medium for making these MIBs available to LAN NetView 1.0 customers. Future versions of the LAN NetView platform will provide more comprehensive SNMP device management capability.

LAN Management Utilities/2 (LMU/2) is a suite of applications for managing OS/2, DOS, LAN Server, and NetWare clients and servers. The application function provided by this product will be intercepted by the LAN NetView family of products over time. However, the LMU/2 product is planned for integration into the LAN NetView family, and in fact has recently changed its name to LAN NetView Management Utilities for OS/2 (LMU).

LAN Network Manager complements the LAN NetView family of products by fulfilling the need for media management on the network. LAN Network Manager can coexist with LAN NetView Manage, contributing valuable function to an overall systems management solution. The 9/92 LAN Network Manager Statement of Direction stated the "goal to provide integrated LAN network management from NetView, NetView/6000, and OS/2 distributed systems management platforms".

Since this paper focuses on the LAN NetView family of products, the positioning aspect is primarily aimed at how it fits in with the other IBM platforms and networking applications.

2.8 Future Directions

As systems management and network management requirements grow, and the trend toward downsizing continues, the LAN NetView product will be expected to respond to these needs. Monitoring and controlling LAN/WAN resources is becoming a high priority issue; one that's evolving more and more toward distributed management. Technology and standards are constantly being developed in this area, and the LAN NetView family of products will continue to take advantage of these improvements. The following subsections give an indication of some of these areas where the LAN NetView products will use future technology and provide additional function.

2.8.1 License Management

IBM has negotiated with Gradient** Technologies, Inc. for use of the NetLS technology to allow license management of OS/2 software. This technology was selected by the Open Software Foundation (OSF) and others as the preferred licensing technology, and is used today in many UNIX** systems including IBM's AIX system.

By porting this technology to the OS/2 environment, it will be possible for OS/2 software to be license-enabled using industry standard technology. License enabling opens the door for many advances in the way OS/2 software is packaged, distributed, and marketed. Keys become the purchased asset rather than the code itself. Concepts like multi-product CD packaging, try-and-buy software, electronic distribution, concurrent usage licensing, free tradeshow samples and many others become far more feasible and attractive alternatives to present methods of merchandising OS/2 software.

In addition to providing a license enabling tool for OS/2 software developers and the runtime environment necessary to support license enabled software, IBM will want to insure that LAN administrators are equipped to manage the enabled software environment with the appropriate LAN NetView product's administration functions.

2.8.2 DME/DCE

As mentioned earlier in this paper, the underlying components of the LAN NetView management framework are based on technology selected by the OSF for the Distributed Management Environment (DME). Since the development of LAN NetView 1.0 was concurrent with the development of DME, a full DME implementation was precluded. As DME components are integrated and this technology matures, it is anticipated that future versions of the LAN NetView products will take advantage of additional DME component development.

It is understood that security is a major issue in systems and network management, and the LAN NetView platform plans to address this by using the security services provided by IBM's OS/2 Distributed Computing Environment (DCE) implementation. DCE will also provide timing services and directory services, as well as RPC (Remote Procedure Call) capability.

2.8.3 DSOM

SOM (System Object Model) is used by the View component of LAN NetView Manage 1.0 as the basis for its GUI. SOM classes are used by the applications to communicate with the View GUI. Applications can also communicate with each other via SOM objects; however, this is only true within a single process. With the addition of DSOM (Distributed System Object Model) SOM objects can communicate across processes and across the network, providing a greater scope of communications for the applications. The LAN NetView product plans to utilize DSOM to take advantage of this efficiency in future releases.

2.9 OS/2 Systems Management Summary

The OS/2 2.x platform provides the necessary operating system capabilities to allow the creation of very powerful systems management applications. At the same time, the power of the OS/2 LAN system has generated the need for these powerful applications. IBM has provided, in the LAN NetView family of products, a suite of these applications to manage many of the IBM resources. The open systems management framework will allow vendors and customers to add management applications and agents to extend the management to other resources.

For the OS/2 Distributed Systems Management environment, the LAN NetView family of products provides the systems management solution required for LAN and WAN-based systems and network management and, in concert with other SystemView platforms, provides the best management solution for the LAN workgroup environment.

The delivery of the initial set of LAN NetView products by IBM and other industry leaders in systems management software begins the realization of the best managed LAN environment in the industry.

Chapter 3. LAN NetView: Framework Overview

This paper discusses the OS/2 distributed systems management framework provided by the IBM LAN NetView Manage and Enabler products. This framework allows management applications to address a wide range of systems management problems utilizing consistent and industry standard interfaces.

This paper is intended for IBM marketing representatives, IBM system engineers, customers and software developers who desire a general understanding of the systems management capabilities provided by the IBM LAN NetView product. It focuses on the LAN NetView framework products and the potential for using them when developing systems management applications.

3.1 Introduction

The IBM LAN NetView family of products provides a framework and applications to implement OS/2-based distributed systems management solutions. The LAN NetView framework utilizes industry standard interfaces and protocols that allow an OS/2 system to manage heterogeneous systems in LAN and WAN environments. An OS/2 system may also be managed by other systems that conform to the same standards.

The LAN NetView family of products is based on systems management standards such as those developed by ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) as part of their work on Open Systems Interconnection (OSI).

The primary purpose of this paper is to provide the reader with an overview of the major components of the LAN NetView framework and how it can be used by management applications to address systems management challenges.

3.2 Directions

In the past, management tools have often grown as extensions to the operating systems of distributed processors. This has led to many unique features and functions of management tools based on the operating system they were derived from. Management services today are either nonexistent, vary in their interfaces, or vary in the types and quality of data gathered. However, the requirement for managing a distributed processor generally does not differ by operating system. To provide an effective environment for the management of distributed systems (that may run different operating systems today or in the future), systems management functions are best made independent of the operating system.

What is needed is a consistent user interface, a consistent set of information and a consistent method for sharing data between management applications.

The directions for distributed systems management are:

- Common graphical end user interface

  The best manner to convey large amounts information quickly is through a graphical interface. An important aspect of a graphical interface is the seamless integration of the underlying management applications. This provides two benefits:

- consistency for the user of the applications and
- consistency for the writers of management applications.

Applications, in this environment, can make high level function calls for graphical display actions rather than have the application itself provide the graphical functions.

- Application platform

  After applications are freed from the details of end user interaction, they can focus on the management functions that need to be performed. Management is divided into different disciplines (problem management, change management, etc.) and applications usually address these disciplines. If the application interfaces are "open", the focus for the application developer can be the application function and not its interface.

- Management Services Framework

  One other area that management applications today must deal with is the variety of ways that management data is collected and processed by devices. The direction of IBM distributed systems management is to adopt an industry standard ("open") management application programming interface (API) that insulates the management application from the network and management protocols supported by the managed system.

- Resource management agents and objects

  In order that management applications can be written independently of the device or system being managed and that new management data may be made available to applications without rewriting management applications, managed data is represented as objects which are accessed through resource manager agents. Again, this provides a layer of insulation between the managing applications and the management data itself.

3.3 SystemView

SystemView is the IBM systems management strategy for planning, coordinating, and operating open, heterogeneous, enterprise-wide information systems. SystemView includes a set of applications which conform to the SystemView structure definitions and integration criteria. The SystemView structure is designed to provide the customer with a consistent user interface, shared data, enhanced automation and increased integration among systems management products. These aspects of SystemView are called dimensions and are respectively named:

- End-Use Dimension

  Provide a consistent user interface.

- Data Dimension

  Shared/consistent data model, definitions, objects.

- Application Dimension

  A managing/managed systems relationship implemented by a set of management services, not integrated into each individual application.

SystemView exploits several key technologies:

- Graphical User Interface
- Relational data (with SQL access)
- Object Oriented Programming
- X\Open Management Protocol API - XMP API

Note that management applications can be written to the XMP/XOM API and be insulated from the management protocol used by a managed system.

SystemView applies to all SAA platforms. The platform to be used is the one most appropriate for a given environment.

3.4 OS/2 and SystemView

OS/2 has three roles in the SystemView structure:

- Platform for the End Use Dimension

  OS/2 will provide the End Use Dimension to the managing platforms. The power of the graphical display will be exploited to provide a consistent user interface to the operators, managers and administrators.

- Managing Platform

  Management applications consistent with the SystemView structure and disciplines will be implemented on OS/2. The LAN NetView platform is the OS/2 implementation of SystemView.

- Managed Workstation

  OS/2 will also be a managed workstation. LAN NetView Enabler provides the OS/2 SystemView platform for being managed.

3.5 Managers and Managed Systems

The concept of managing/managed systems is fundamental to distributed systems management. The Open Systems Interconnect (OSI) project defines the roles of managing and managed systems. These roles are summarized below:

- A managing system is responsible for the management of other systems.
- Managing systems components
  - End user interface or automated operator

    The managing system may have a user interface to display information gathered by the management applications. If an operator is not available or not required, automation can be used to analyze the data collected and act on anticipated events.

  - Management process application

    A managing system contains some managing process. This is an application that processes the management data received from the managed devices. A managing system can also be a managed system. Information gathered at one manager can be forwarded to another in either a hierarchical or peer relationship.

- Managed systems contain agents that provide a linkage between the objects to be managed and the transport to the manager.

  A managed system is responsible to provide information about itself to a managing system.

- Agent

  The agent responds to commands from the manager and collects requested data concerning objects in the managed device. Agents also send unsolicited information to the manager in certain conditions.

- Managed objects

  Objects are status, characteristics, and data about some specific aspect of the managed device. An object can represent hardware or software information. These objects are collectively referred to as a management information base (MIB).

IBM's LAN NetView product will conform to this architecture and be the strategic distributed systems management offering in the OS/2 environment.

3.6 Management Protocols

Due to the evolving nature of the Open System Interconnect (OSI) system management standards and the rapid acceptance of the SNMP approach to network management, a multi-management protocol design has been chosen as the most viable design to address the needs of the marketplace in the 90's. So, the LAN NetView platform's architecture currently supports two different sets of protocols and services to be used between managing and managed systems:

- CMIP

  CMIP (Common Management Information Protocol) is defined by ISO and is implemented in several environments. These environments include TCP/IP and 802.2 LLC. When used in these environments it is often referred to as CMOT (CMIP over TCP/IP) and CMOL (CMIP over LLC).

  The CMIP protocol provides a set of primitives for accessing management information through a set of services called CMIS (Common Management Information Services). These services include the capability to GET or SET specific attributes of managed objects, CREATE or DELETE instances of managed objects, request a managed object perform a specific ACTION, or emit a notification from a managed object about a specific EVENT that may be of interest to a managing system.

- SNMP

  SNMP (Simple Network Management Protocol) is a simple protocol by which management information for a network element may be inspected or altered by logically remote users. It is a transaction-oriented protocol that allows network elements to be queried directly.

  SNMP provides a means for managing an Internet environment. Implicit in the SNMP architectural model is a collection of network management stations and network elements, such as gateways, routers, bridges and hosts. These elements act as servers and contain management agents which perform the network management functions requested by the network management stations. The network management stations act as clients; they run the management applications which monitor and control network elements.

  SNMP provides a means of communicating between the network management stations and the agents in the network elements to send and receive information about network resources.

Like CMIP, SNMP also provides GET and SET functions for accessing the attributes of managed objects.

Though the two management protocols provide overlap in some functional areas, they were designed with different objectives. The OSI model and CMIP were intended to provide a complete solution and in general provide a richer set of capabilities.

SNMP was originally designed to be simple and small and to provide the capability for simple prototyping of management solutions. Over time, SNMP has become more powerful, though there are still capabilities that are provided through CMIP that are not as easily implemented in an SNMP environment.

The major difference is that SNMP uses a polling model with lots of function in the management applications and very little in the agent. CMIP, on the other hand, uses an event-driven model with more function in the agent and potentially less in the application. CMIP, in theory, results in lower network traffic.

The LAN NetView platform provides support for CMOT, CMOL and SNMP protocols. However, due to the capabilities inherent in the protocols themselves and the design of the LAN NetView agents and applications, one may find that the CMOT and CMOL environments provide more function than using the LAN NetView products in a SNMP environment.

3.7 The LAN NetView Family of Products

The LAN NetView family of products includes support for both managing and managed systems. The environment is depicted in Figure 1 on page 56.

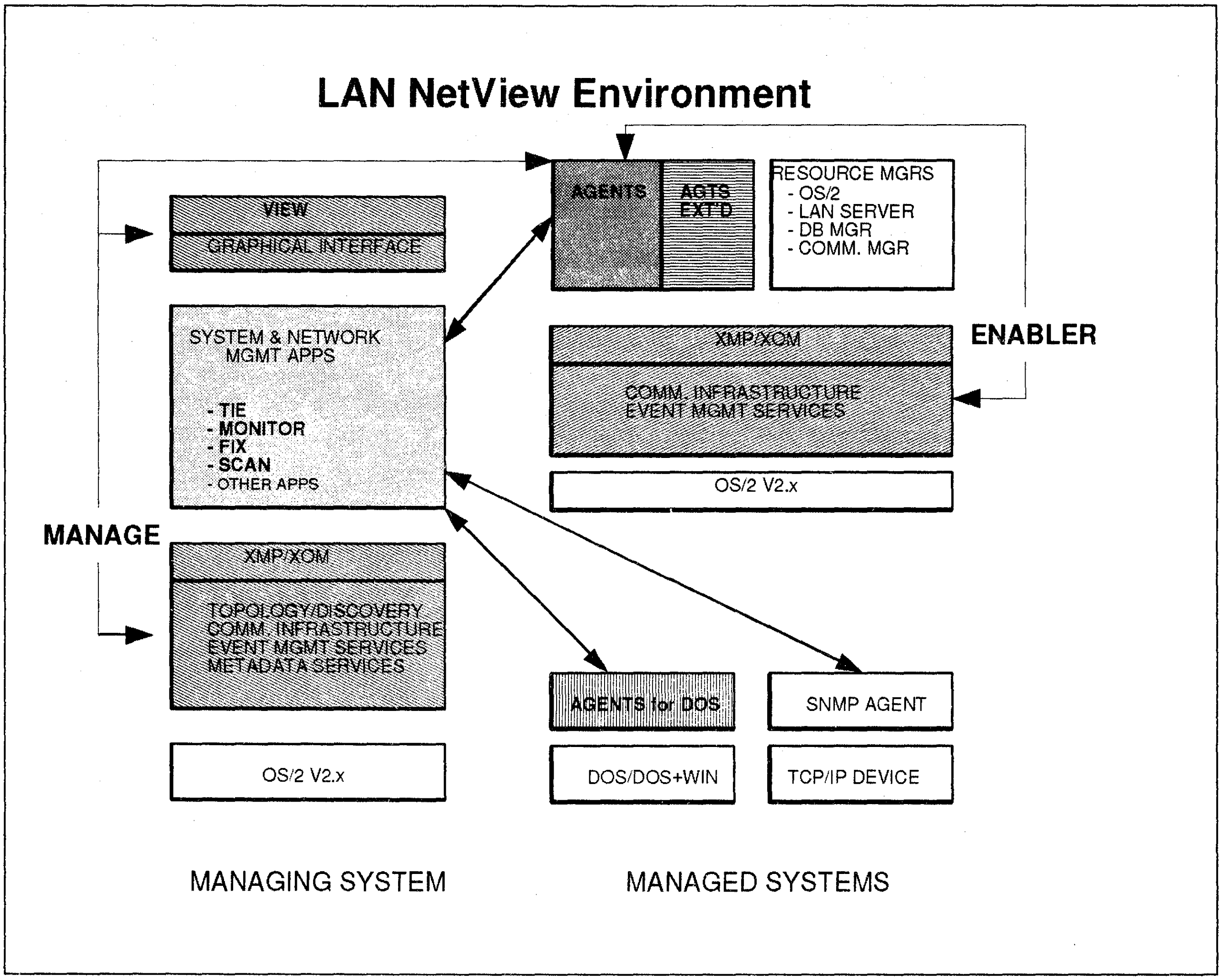


Figure 1. LAN NetView Environment

The following sections will briefly describe the products that comprise the LAN NetView family of products and the roles they play.

3.7.1 LAN NetView Manage

LAN NetView Manage 1.0 provides the core functions required by a managing system. These include a communications infrastructure, event management, metadata and topology/discovery services. The LAN NetView Manage product provides the industry standard X/Open Management API's (XMP) and the X/Open OSI-Abstract-Data Manipulation API's (XOM) for the development of management applications. By utilizing the XMP API's, manage applications can be written to utilize either (or both) SNMP or CMOT/CMOL protocols.

In addition, LAN NetView Manage 1.0 includes the View component (hereafter referred to simply as View) which provides a graphical interface to the LAN NetView platform. Developers may use the View programming interfaces to deliver a consistent look and feel to their management applications.

View will automatically provide for easy navigation through the network hierarchy. It also provides services to allow displaying of management data in progressive layers of detail.

LAN NetView Manage 1.0 also provides the OS/2 and LAN Requester agents. These agents are normally installed on a managed system. In some cases, a managing system may itself be managed by another managing system. Also, a managing system may include itself as a managed system when gathering management data. In either of these cases, these agents may also be installed along with LAN NetView Manage 1.0 on a managing system.

Two application level functions are also provided with Manage: the Request Manager and the Remote Command Line Interface (RCLI). The Request Manager allows the system administrator to access function in IBM agents and other CMIP and SNMP agents. The Request Manager allows the user to query attribute values of objects represented by agents as well as to set selected attribute values. This utility can be used to provide management functions that are not available by using any of the available applications.

The RCLI allows the LAN administrator to issue commands from a managing workstation to be executed on a managed workstation. This function will be useful for such things as starting and stopping OS/2 applications, changing configuration values in OS/2 applications and querying the status of an OS/2 system.

3.7.2 LAN NetView Enabler

LAN NetView Enabler 1.0 provides the managed system platform on OS/2.x-based systems. It also provides the XMP/XOM programming interfaces for the development of management agents which will interact with applications on the managing system.

The LAN NetView Enabler product also provides the OS/2 and LAN Requester agents that will allow a managing system to manage those resources controlled by the base operating system or the LAN Requester.

3.7.3 LAN NetView Agents for DOS

This product provides the agents required for IBM DOS V5.0/6.1, Microsoft DOS V5.0/6.0, and Microsoft Windows V3.0/3.1. IBM's managing applications require these agents to be installed on any DOS/Windows systems that are to be managed.

3.7.4 LAN NetView Agents Extended

The LAN NetView Agents Extended product provides the agent support for the LAN Server V3.0, Database Manager and Communications Manager subsystems.

3.7.5 LAN NetView Applications

The following management applications have also been announced by IBM. These applications run on a managing system (requiring LAN NetView Manage 1.0) and will utilize the agents on the managed systems.

LAN NetView Fix - is a general purpose event handling application that enables software products to automate their problem determination procedures in a manner that can be tailored and extended by customers. The Fix product receives and processes both CMIP events and SNMP traps that are emitted from managed machines on the network. Users specify the actions that the Fix program is to take based on the events received. Both LAN NetView Fix 1.0-supplied and user-written actions can be specified.

LAN NetView Monitor - provides automated performance management through the use of user-defined policies that specify the resource data to be collected, collection schedules, threshold levels and actions, and data transfer times.

Data may be collected and stored in a relational database. A graphical display of data collected can be presented as it is collected, or from data previously stored in the database.

LAN NetView Tie - provides a mechanism for the filtering and transmission of notifications emitted on the LAN, to a NetView host. LAN NetView Tie 1.0 can register to receive both OSI-alarm and non-alarm notifications. The LAN NetView Tie program converts OSI alarm notifications to SNA alerts or resolutions. Non-alarm CMIP events are wrapped in a SNA/MS major vector along with other related information and sent to the host. A receiving application must be available to unwrap and parse this information.

LAN NetView Tie 1.0 can be configured by a LAN NetView administrator or through NetView RUNCMD's at a host based NetView console.

LAN NetView Scan - is currently available through a Beta program. The announcement of this product's availability will be determined by the experience of the Beta participants and the feedback they provide to IBM on the LAN NetView Scan product's function and usability.

The LAN NetView Scan product provides function for configuration and inventory management of LAN-attached workstations running OS/2, DOS and DOS with Windows.

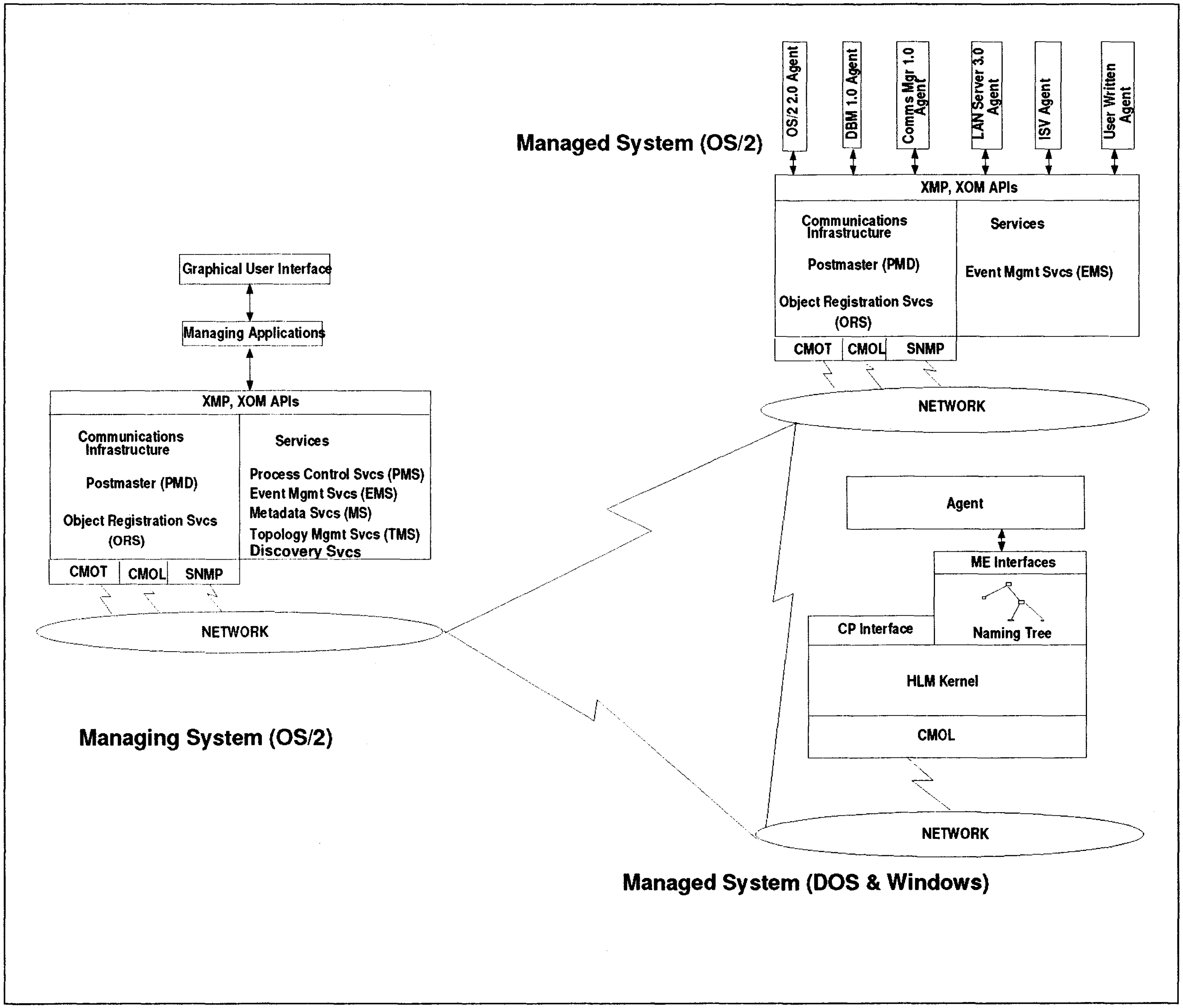
The LAN NetView Scan product can collect workstation Vital Product Data as well as monitor and collect workstation files. The information and/or files that are gathered are stored in a centrally-located SQL database.

The LAN NetView Scan product also provides a provides a command scheduler. At regularly scheduled times, it will run programs or commands at selected OS/2 workstations to perform such tasks as software inventory, system backup, virus checking, system diagnostics and report generation.

3.7.6 LAN NetView Framework

The core of the LAN NetView environment is the framework provided by both the LAN NetView Manage and LAN NetView Enabler products.

The following diagram highlights the components of the framework in a LAN NetView environment. The discussion after the diagram will provide more detail on the roles that each component plays.


Managed System (OS/2)
OS/2 2.0 Agent
DBM 1.0 Agent
Comms Mgr 1.0 Agent
LAN Server 3.0 Agent
ISV Agent
User Written Agent
XMP, XOM APIs
Communications Infrastructure
Postmaster (PMD)
Object Registration Svcs (ORS)
Services
Event Mgmt Svcs (EMS)
CMOT
CMOL
SNMP
NETWORK
Graphical User Interface
Managing Applications
XMP, XOM APIs
Communications Infrastructure
Postmaster (PMD)
Object Registration Svcs (ORS)
Services
Process Control Svcs (PMS)
Event Mgmt Svcs (EMS)
Metadata Svcs (MS)
Topology Mgmt Svcs (TMS)
Discovery Svcs
CMOT
CMOL
SNMP
NETWORK
Managing System (OS/2)
Agent
ME Interfaces
Naming Tree
CP Interface
HLM Kernel
CMOL
NETWORK
Managed System (DOS & Windows)

3.8 The LAN NetView Framework Overview

The LAN NetView family of products is based on a series of industry standards as mentioned previously. This family of products provides systems management applications and a framework. This framework provides an enabling platform for IBM applications as well as applications written by vendors or customers.

3.8.1 Framework Components

The LAN NetView framework has several core components. The LAN NetView Manage product is a superset of the LAN NetView Enabler product. We will discuss the components within the LAN NetView Manage product context and will highlight any differences between the two products.

It should be noted that only agents can run on LAN NetView Enabler. More specifically, only applications performing the agent role are allowed on Enabler.

Figure 2 on page 61 provides a high level schematic of a managing system.

The diagram shows three layers of a LAN NetView managing system:

- The graphical user interface (View) presents the user with a graphical representation of the managed components. The user accesses the managing applications from this interface. View is part of the LAN NetView Manage product.
- The managing applications can be provided by IBM, third party vendors, or they can be user-written. These managing applications can utilize View APIs and/or the XOM/XMP interfaces to the managing framework. These applications are typically sold separately, however some mini-applications (also called applets), such as the Request Manager, and Remote Command Line Interface, are packaged with the LAN NetView Manage product.
- The LAN NetView Manage framework can be broadly divided into two categories:
  - The communications infrastructure facilitates the exchange of messages between the managing applications and the managed systems.
  - Management services that support the development of managing applications.

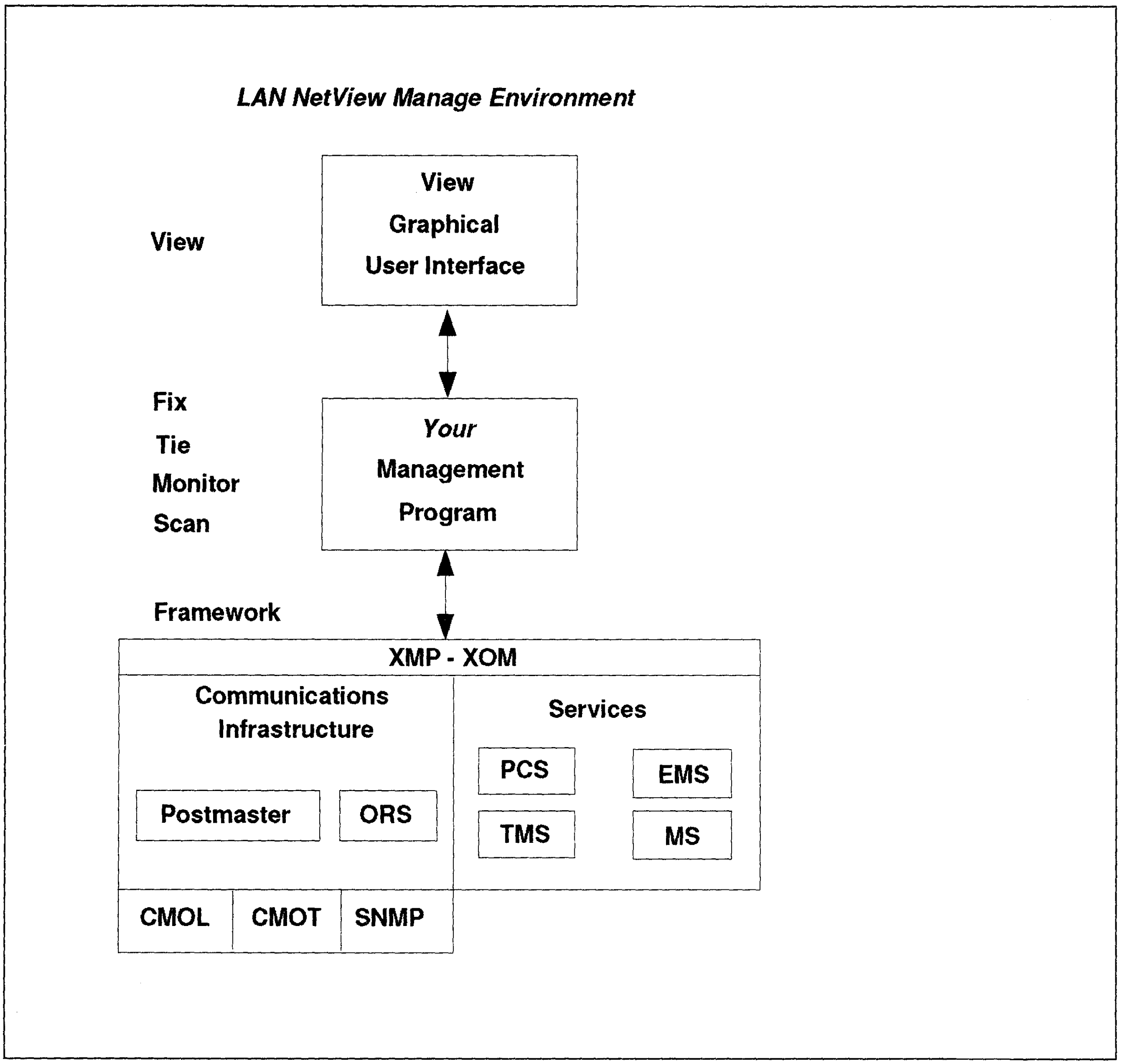


Figure 2. LAN NetView Manage

The following sections will discuss each of these components in some detail and how they are typically used to provide systems management solutions.

3.8.1.1 Communications Infrastructure - Postmaster and ORS

The LAN NetView platform's communications infrastructure is engaged by any application using the XMP programming interface. The LAN NetView platform supports two different sets of management protocols between managed and managing systems, CMIP and SNMP. The XMP interface provides a generic programming interface and is used regardless of the specific protocol that will ultimately be used for communications between managed and managing systems.

The Communications Infrastructure consists of two parts:

- Postmaster
- Object Registration Service (ORS)

Together they provide the services required for messages to be routed between managing and managed systems:

- Message routing:

  The communications infrastructure is the mechanism by which messages are exchanged.

  - Object location transparent approach

    By providing location transparency, managers (management applications) may gain access to managed objects without explicit knowledge of the physical location (address) of the managed system. It is not necessary to separately determine the address of the agent and supply that address to each request.

    For location transparency to work, the object and its agent must be registered with the communications infrastructure via the Object Registration Services (ORS) (see below).

    If an application specifies only the class and name of an object instance, the communications infrastructure uses the database created by ORS to route the requested operation to the agent responsible for that object instance.

  - Non-location transparent approach

    An application may also use a non-transparent approach, where an address of a specific agent is specified for each request.

- Association Management

  For CMOT/CMOL requests, the communications infrastructure provides transparent control and initiation of the connection-oriented associations.

- Automatic Retries

  For SNMP requests using the connectionless User Datagram Protocol (UDP), the communications infrastructure hides the details of managing timeouts and retries.

Postmaster: The postmaster is a message switch which directs management information between managers and agents. It determines its routing action either from user specified addresses or from routing tables configured through the Object Registration Services (ORS).

The postmaster also hides many of the differences between CMOT/CMOL and SNMP requests. This allows applications to be written in a more generic fashion and to function independent of the management protocol used in a specific environment.

Note: Though many of the details are hidden from the programmer, there are still functional differences between CMOT/CMOL and SNMP which the application designer must take into account. The differences between a CMOT and a CMOL environment are completely hidden from the programmer by the postmaster.

Object Registration Services: In order for the postmaster to function in the location transparency mode, it must have access to a directory which will be used to map object instances to specific physical addresses in the network. These 'directory services' are provided through the Object Registration Services component.

Object registration is the process of registering agents and associated objects with the ORS database. The ORS creates and maintains a global directory of agents, their locations, the objects each agent manages and the protocol that should be used to communicate with the specific agent. To operate in location transparency mode, each agent that will be accessed from a managing application must be registered with the ORS.

ORS permits dynamic modification of object-registration information.

An ORS database resides on both managing and managed systems. On a managing system, it is used to identify the address of the managed system a specific request is to be routed to and the protocol to be used. On a managed system it is used to determine which agent controls the object being addressed so that the request can be passed to the appropriate agent.

Communications Infrastructure Summary: Managing applications use the X/Open Management API's (XMP/XOM) to issue requests against object instances. The Postmaster daemon handles these requests and will typically use the ORS database to identify the appropriate protocol and destination address to use. Postmaster services then will ensure that any associations between managing and managed systems that are required are formed, and will send the request appropriately. (In an SNMP environment, there are no associations, as transmissions are sent as connectionless datagrams, however the postmaster will use timeouts and retries to provide similar assurance of message delivery.)

On a managing system the postmaster tracks all outstanding requests and correlates incoming responses to ensure they are passed to the appropriate managing application.

In a managed system, the postmaster will receive in-coming requests and use ORS if necessary in order to route the request to the agent controlling the object instance being addressed. Any responses from the agent will be routed back to the managing system that originated the request.

3.8.1.2 Event Management Services

In a distributed systems management environment it is important that individual systems be capable of notifying managing systems of key events that have occurred. Examples of the types of events for which agents may need to generate notifications are:

- Problems have occurred
- Performance or capacity thresholds have been reached
- Critical files have been altered
- State/Status of a system or subsystem has changed

Efficiently handling events in a distributed environment can be difficult. Some of the issues involved include:

- Knowing to which managing system(s) notifications should be sent.

  In many cases, different managing systems may run management programs addressing different disciplines such as performance management, problem management, etc. There needs to be a way of routing the notifications to the appropriate managing system.

- Network traffic load due to a large number of notifications.

  The requirement is to send only those notifications on the network that managing applications are prepared to handle.

- Agents should not be aware of managing systems and the events they are interested in.

  Agents should only be concerned with the particular resource manager they interface with, and should not be concerned with maintaining routing tables and lists of managing systems to which notifications need to be sent.

In a LAN NetView environment, Event Management Services (EMS) provides the facilities to address the above requirements. EMS resides on both managing and managed systems and is composed of Event Sieves, an Event Sieve Agent and an Event Log Agent.

3.8.1.3 Event Sieves and the Event Sieve Agent

Event sieves allow LAN NetView to minimize network traffic while ensuring that all managing applications receive the event notifications they require. An event sieve is like a filter which will examine each event generated by an agent and route that event to the appropriate managing system and/or management application.

Event sieves exist on both managing and managed systems. On a managed system, the event sieve processes all events generated by any agent that is running on the managed system. Any events that pass through the sieve are then forwarded on to any managing system that has registered for such events. In other words, EMS on the managed system filters the events that have been generated and routes them to the appropriate managing systems.

On a managing system, EMS receives all notifications that arrive at that station and using event sieves, pass the event notification to whichever managing applications are appropriate. An event sieve on a managing system filters all received notifications and routes them to only those managing applications that have registered for the particular event type.

Event sieves are managed by the event sieve agent. Also, the event sieve agent handles the routing of events once they have passed through an event sieve.

There are two types of event sieves, Static and Dynamic. Static sieves require no programming, but they only exist on managed systems and provide no filtering of event types. Static sieves allow a LAN NetView administrator to configure a managed system such that ALL events generated will be forwarded to a specific set of managing systems.

Dynamic sieves are created via XMP programs and can contain very specific filters in order to limit the network traffic. Dynamic sieves can be created on managed systems as well as managing systems. Dynamic sieves are the only type of sieve that can pass received event notifications to management applications. Therefore, dynamic sieves will need to be generated by any managing application wishing to receive event notifications.

Typically, management applications will generate dynamic sieves on both the managing system and on managed systems. By doing this, the managing application can ensure that all events of interest are passed back to itself.

Filtering Event Reports: The volume and diversity of events occurring in a distributed system can be very large, making it difficult to separate significant information from noise. A management program can filter event reports; that is, a management program can indicate which event reports are to be forwarded or delivered to it. Filtering can minimize network traffic by preventing the forwarding and delivery of unneeded event reports.

A filter is specified as part of the event sieve.

In order to limit the network traffic, an event report can be filtered using any of the following event attributes and their associated values:

- Object-class
- Object-instance
- Event-time
- Event-type
- SNMP-trap
- SNMP-specific-trap

In addition, event reports can be filtered by frequency; that is, a filter could specify that a single event report should be forwarded only if a certain number of those events are generated in a particular amount of time. For example, a filter could specify to pass only one protocol retry event report in each one minute period during which three or more protocol retry events occurred.

Conversely, a filter could specify that event reports not be forwarded when the number of generated event reports exceed a specified number in a particular time period. For example, a filter could inhibit the forwarding of unreachable node event reports when more than five are generated per second thus avoiding flooding the system with unnecessary event reports.

To create more complex filters, you can combine simple filters with the

- AND
- OR
- NOT

operators.

3.8.1.4 Event Log Agent

Event Management Services also provides an Event Log Agent that allows default logging of all events received at a particular system.

By having events logged, management applications can retrieve event information that may have occurred before the managing application was started.

3.8.1.5 EMS Summary

The LAN NetView framework provides a rich set of services for handling events generated by agents within the network. These services allow application programmers flexibility in working with only specific types of events for which they are designed. At the same time, network traffic can be kept to a minimum as events for which no managing application has an interest can be suppressed.

The Event Log Agent provides additional capabilities for generic logging of events. The event log can be accessed by management applications in order to gain information about events that had been generated before the management application became active.

3.8.1.6 Topology/Discovery

An important part of distributed systems management is to understand the topology of the network(s) being managed. Understanding the topology includes understanding the physical makeup of the networks, the logical relationships between systems on the network and the identification of devices within the network.

The objectives of the Topology Management Services component of LAN NetView Manage is to discover the physical systems on the network and to collect data about these systems to assist in systems management functions.

Since the make-up of a network is very dynamic, it is not enough to discover the systems, but changes in the network must also be monitored. Topology Management Services will monitor discovered systems for changes in their state/status.

Topology Management Services will gather data about the physical systems as well as their logical relationship to one another.

Topology Management Services is made up of three primary components:

- Topology Agent
- Discovery Processes
- Discovery Database

The discovery processes work independently to discover and identify the systems that make up a network. In LAN NetView the discovery processes will discover systems that are using the TCP/IP, HLM and IPX protocols. In addition, user written discovery processes may be added to discover systems that are utilizing other protocols.

The topology agent manages the discovery database. It collects information from the discovery processes and adds it to the discovery database. In addition, the topology agent will attempt to identify individual systems that may have been discovered by different discovery processes and consolidate that information. For instance, a single OS/2 system could be using both the HLM (CMOL) stack and TCP/IP in which case, both discovery processes will have reported finding the same physical system.

The topology agent also allows management applications access to the information in the discovery database. The View component of LAN NetView Manage constantly queries the discovery database in order to display the current topology of the managed network.

Once physical devices have been discovered on the network, TMS attempts to gather as much information about the individual devices as possible. How much information can be gathered will be determined by the agents that are installed on the discovered systems.

If the LAN Netview agents are installed on the discovered system, then the discovery database will contain information regarding the resources attached to the system. This can include both hardware and software information. Management applications can then query the system directly to obtain even more information about the individual resources installed.

3.8.1.7 TMS Summary

Topology Management Services provide discovery processes for SNMP, CMOT, CMOL and IPX devices. In addition, if LAN NetView agents are installed on the discovered system additional detail will be gathered and stored in the discovery database. An application can then query the discovery database through the topology agent in order to obtain information about the discovered devices. The application can then access the device directly to obtain information above and beyond that which the discovery database contains.

View, the graphical front end to the LAN NetView Manage product utilizes the topology management services in order to display the current topology of the managed network.

3.8.1.8 Metadata Services

A Management Information Base (MIB) is an abstract view of all the object classes in the network that can be managed. Many application designers would find it desirable to be able to make their applications as generic as possible so that the same program could be used to manage new objects that may become part of the network. For instance, a management application that is designed to manage all of the printers in the network, should be flexible enough to handle new types of printers that may have advanced functions beyond those available today.

In order to provide this capability to LAN NetView application developers, the MIB information is controlled by a set of services called Metadata Services.

These services allow new object types to be defined to the MIB and to allow applications to query the MIB. Thus, applications can dynamically build management requests to be passed onto a target object.

Metadata Services is made up of three components.

- Metadata Manager - Primarily consists of a compiler (METACOMP.EXE) which takes ASCII files containing the definitions of managed object classes and places the object definitions in the metadata database. These ASCII files must conform to the ISO 10165-4: Guidelines for the Definition of Managed Objects (GDMO) standard.
- Metadata Agent - Controls the access to the metadata database. Requests to access the metadata database are routed to the metadata agent which performs the requested function/action.
- Metadata Database - This is the LAN NetView platform's MIB. It is accessed through the metadata agent.

Metadata services provides the capability to write more generic applications (such as browsers). It allows management programs to learn about managed objects at run time.

Without rewriting applications new types of objects can be managed by adding the information to the metadata database.

3.8.1.9 Process Control Services

As we have seen, there are many separate services that are provided by the LAN NetView framework. Many of these services are dependent on one another. It is important that when starting a LAN NetView system, the proper pre-requisite and co-requisite services are started in the proper order to ensure that all services function properly. The LAN NetView platform services used to achieve this synchronization are the Process Control Services.

Using the process control services a user can:

- Control when an agent or service is started.
- Ensure pre-requisite services are started.
- Define the startup parameters that are passed to an agent/service.
- Define a timeout value that is used when an agent experiences a startup failure.

Process control services allows a managing or managed system with LAN NetView Manage 1.0 or LAN NetView Enabler 1.0 respectively, to be started in a controlled manner with no user interaction required.

This allows LAN NetView platform processes to be initiated in a way that is least disruptive to the users.

3.8.1.10 View Overview

Though the View component of LAN NetView Manage 1.0 is not part of the *framework* which we have been discussing, it is a critical component of the managing system. View provides the user interface for the LAN administrator who will be managing a network.

View provides three primary functions on a LAN NetView Manage 1.0 system.

- It provides a GUI front end that displays the current network topology and allows the user to navigate through the various levels of detail concerning that topology.
- It provides a platform for the integration of various LAN NetView-based applications.
- It provides access to a programming library of System Object Model (SOM) based objects that can be used to build applications on the LAN NetView platform that will use the GUI interface.

The GUI provided by View allows the user to navigate through a hierarchy of containers in order to select the object of interest. The user can then select from a list of actions on this object such as invoking various management applications.

The View GUI presents a three tiered presentation of the distributed environment. These three tiers are the:

- Management Collection - which is a logical grouping of related systems.
- Systems - which are devices associated with a physical address on the LAN.
- Resources - which define objects contained within the systems.

View provides the capability to define background maps to be used with management collections. The facilities are provided to automatically position systems on the map to correspond to their actual location based on information retrieved from the system itself.

3.8.2 Management Utilities

The LAN NetView Manage product also provides two 'mini-applications': Request Manager and Remote Command Line Interface (RCLI).

The Request manager allows a LAN administrator to browse and in some cases set attributes associated with objects on managed systems. In addition, the Request manager allows for the creation of event sieves on managed systems. The Request Manager can be used to interact with the IBM agents or other CMIP and SNMP agents located in the network.

The Remote Command Line Interface (RCLI) allows a LAN administrator at a managing system to execute commands on managed systems that contain the LAN NetView system agent. OS/2 commands and REXX procedures can be initiated providing a very powerful environment for querying and modifying the environment of a managed system, especially in areas that the current agents do not provide access to.

3.9 LAN NetView Solutions Summary

This chapter will summarize the topics discussed in the previous chapters from a "solutions" point of view.

The LAN NetView framework is a series of services that run on managing and managed systems. These services provide the functions required to implement management applications that:

- can be generic or specific in nature
- can be written independent of the management protocols ultimately used within the network
- do not have to be concerned about physical addressing of systems in the network
- can register for, and respond to specific notifications of events occurring on managed systems
- can provide a common graphical look and feel, and can be initiated through interaction with a graphical view of the network as it currently exists

The communications infrastructure provides postmaster services for handling the routing of requests between systems. The requests need only specify object instances, since the postmaster can use the Object Registration Services database to identify the physical location (address) of the object. The postmaster also uses ORS to identify the protocol stack to use when communicating with a specific object instance. This allows the LAN NetView programmer to address object instances at a high level and not be concerned with the lower level communications-related details that otherwise might be required in writing a distributed management application.

The communications infrastructure as described above will be used by most managing applications that will be requesting information about, or actions to be performed by, a managed system. This could include querying systems for vital product data, creating or deleting specific object instances, requesting a particular action to be performed by the managed object, etc.

Event Management Services provides the facilities to control the routing of event notifications to the appropriate managing system(s) and application(s). By utilizing event sieves, network traffic and the load on a managing system can be reduced as only those events of interest will be transmitted. The LAN NetView agents provide a rich set of event notifications. The events supported by these agents address all areas of distributed systems management such as problem management, performance management, inventory, configuration and change management.

The topology/discovery services provide the managing system with a view of the network as it currently exists. The topology information is constantly updated as systems enter and leave the network. In addition, information regarding the agents and resource managers available on individual systems is also dynamically updated. The information gathered by topology/discovery is primarily used by the View component to give the user of the managing system(s) a graphical view of the network and the status of the devices on the network.

By selecting a system or group of systems from the View graphical display, the user can invoke management applications to query, perform operations or otherwise manage the selected system(s).

Metadata services is used by applications to dynamically query the MIB (or the definitions of the managed object classes). Applications using this service dynamically build requests to interact with object classes that are defined in the MIB. The value of this is that applications written with this in mind, will be able to manage new or changed object types that have been defined after the application was written. A prime example of an application that would use metadata services is a MIB browser. Such an application could initially use Topology/Discovery services to identify the systems in the network. It would then query the systems to identify agents and objects supported by those agents that are installed on the individual systems. Finally, it could use the metadata services in order to correctly query the objects on those systems in order to display detailed system information to the user.

The final set of services we discussed were the process control services. These services provide support to easily start/stop LAN NetView platform processes while ensuring that all pre-requisite and co-requisite services are also started. These services are key to allowing managed systems to be properly initialized without requiring intervention by the end user.

3.10 Summary

In summary, the LAN NetView framework provides several key services that make it easier to develop distributed systems management applications. These services span the functions required to implement applications addressing all systems management disciplines.

The LAN NetView framework provides the core services that are used by applications developed on the LAN NetView platform, on both the managing and managed systems products (LAN NetView Manage 1.0 and LAN NetView Enabler 1.0). This layer of services allows new applications to take advantage of currently installed agents. It also allows currently installed applications to address new agents that may represent new resources installed on managed systems.

Part 2. Systems Administration

Chapter 4. Planning a Management Scheme Using the IBM LAN NetView Family of Products

In setting up a management scheme using the IBM LAN NetView family of systems management products, effective placement of the management entities is important. This paper classifies the major types of distributed LAN system configurations where the IBM LAN NetView family of products is useful. For each configuration type, the issues involved in placing IBM LAN NetView products is discussed.

4.1 Introduction

The proliferation of workstations has changed many aspects of how you conduct your business. This emergence of distributed LAN systems has complicated how you provide and maintain the information services needed to support the new business procedures.

To establish control over your distributed LAN system, you need to revise your management scheme. The IBM LAN NetView family of products provides a rich base on which to build this revised scheme. With a focus on the information services provided by the distributed LAN system, the IBM LAN NetView Manage and IBM LAN NetView Enabler products provide the basic functions needed to manage your system of OS/2 workstations and supporting communications gear. The IBM LAN NetView Agents Extended product fortifies the management of those workstations with the IBM Communications Manager/2, IBM Database Manager/2, or IBM LAN Server products. The IBM LAN NetView Agents for DOS product brings DOS workstations into your management scheme. These products provide the underlying "instrumentation" and support services that allow managing applications to be effectively added.

Atop this base, managing applications provide the functions you need to implement your management scheme. These applications range along a spectrum from those narrowly-focused on particular resource types but providing extensive function to those that provide consistent yet limited function across a broad range of resource types. These applications are available from IBM and from Independent Software Vendors (ISVs).

IBM supplies applications that emphasize consistency in the management of different resources. This focus is based on desiring integrated management of all of the components of a distributed LAN system, thereby simplifying the management scheme that you have to construct. The IBM LAN NetView Monitor product addresses performance measurement and reporting for resources on OS/2 workstations. The IBM LAN NetView Fix product handles trouble reports and gives you a means to automatically react to trouble when it occurs. The IBM LAN NetView Tie product helps connect the management scheme required for your distributed LAN system with an enterprise-wide management scheme based on the NetView product on the mainframe. The IBM LAN NetView Management Utilities product provides a migration path for users of the LAN Management Utilities/2 product into a scheme based on the IBM LAN NetView family of products.

ISVs are also delivering management function atop IBM LAN NetView products. Novell provides an application that integrates the management of Netware servers with the IBM LAN NetView management functions. Protools is providing an application for monitoring the performance of LAN media. Shany, Inc. is providing an alert management application, while Microcom is providing a set of applications for managing DOS and OS/2 workstations.

These products, when properly deployed in your distributed LAN system, will allow you to implement your management scheme effectively. This translates into satisfied users of your distributed LAN system.

4.2 Deployment Considerations

A management scheme may be described as the interaction of four pieces:

Policies and Procedures
The definition of how you wish to manage your distributed LAN system.

LAN Administrators
The humans involved in carrying out the procedures.

Managing functions
The software and hardware that assist the LAN administrator in carrying out his or her responsibilities.

Instrumentation
The software and hardware that provide the means with which the managing functions understand the distributed LAN system, and effect changes to it.

To create an effective management scheme, you should understand the focus of your policies and procedures, the number and location of your LAN administrators, and the characteristics of your distributed LAN system relative to these pieces. The focus of your policies and procedures determine which particular LAN Administrator skills are needed, what managing functions must be used, and where and how the components of the distributed LAN system are instrumented. The number of LAN administrators and their geographic location dictate where to place managing functions and may influence the network protocol(s) used in your distributed LAN system. The networking protocols in use throughout the distributed LAN system also influence the placement of managing functions. Of course, the size of your distributed LAN system affects where and how your LAN administrators and the managing functions are deployed.

Four basic distributed LAN system configurations emerge when considering the factors described above. These types are Classic LAN, Massive LAN, Branch Office, and Remote Service. They may be distinguished based on size, organization served, LAN administration organization, and management emphasis. Table 1 on page 77 outlines the differences between the configuration types. These types, and the options for using the IBM LAN NetView family of products with these types, are discussed in the following sections.

Configuration Type	Size	Organization Served	LAN Administration Organization	Management Emphasis
Classic LAN	Small-to-Medium	Single	Same as Served	Reactive
Massive LAN	Medium-to-Large	Multiple, Diverse	Common, Shared	Reactive, Proactive
Branch Office	Multiple Small-to-Medium	Multiple, Replicated	Common, Shared	Reactive
Remote Service	Multiple Small	Multiple, Unrelated	Common, Outsourced	Proactive

Table 1. Distributed LAN System Configuration Types

4.3 Classic LAN

The Classic LAN configuration is a collection of workstations on a token-ring LAN with a small number of segments. Usually, this configuration supports a single organization or a tightly-interrelated group of departments within an organization. The number of workstations range from a few to several hundred.

LAN administration usually is handled by the organization served by this distributed LAN system. Consequently, whoever gets the LAN administration assignment has other assignments that will take priority until a problem occurs. The management emphasis becomes reactive, to understand the conditions surrounding the problem once it occurs and to resolve the problem quickly. Policies or procedures established for the management scheme reflect this emphasis.

The networking protocol used in this configuration could be NetBIOS, APPN, TCP/IP, AppleTalk**, or IPX**. Fortunately, a common denominator of these protocols in this configuration is Logical Link Control (LLC). The IBM LAN NetView family of products supports a management protocol called CMIP over LLC (CMOL) that provides a common means to move information between the instrumentation and the managing functions.

This configuration can be managed by a single IBM LAN NetView Manage 1.0 workstation, with IBM LAN NetView Enabler 1.0 and IBM LAN NetView Agents Extended 1.0 (if needed) installed on each user's OS/2 workstation or IBM LAN NetView Agents for DOS 1.0 installed on each user's DOS workstation. IBM LAN NetView Fix 1.0 or another trouble report application would be used to monitor automatically for problems. When a problem occurs, the LAN administrator would be notified either by the user seeing the problem or by the IBM LAN NetView Fix application. Using the View component of IBM LAN NetView Manage 1.0 in combination with the Request Manager component, the LAN administrator could determine the state of the user's workstation and issue requests to correct the problem. If performance degradation is the problem, the LAN administrator could use the IBM LAN NetView Monitor application to collect information on the nature of the load on the workstation. This data could then be used to alleviate the bottlenecks.

Due to the size of this distributed LAN system configuration, the need for multiple managing workstations is unlikely. However, nothing precludes you from installing additional managing workstations if the number and location of your LAN administrators demands more.

4.4 Massive LAN

The Massive LAN configuration is the larger cousin of the Classic LAN configuration. The distinction between these two types is complexity. For the Massive LAN, there are more workstations on more LAN segments. The LAN segments may be geographically-dispersed and bridged across Wide Area Networks (WANs). Such a distributed LAN system is usually supported by a common department and used by multiple departments across the organization. The number of workstations may range into the tens of thousands.

The department supporting the distributed LAN system has the LAN administration responsibilities. Since these people are devoted to LAN administration, they have the opportunity to analyze the distributed LAN system and spot potential problems before the users are affected. However, when problems occur, the primary responsibility of the LAN administrator continues to be understanding the problem and correcting it quickly. Thus, the management emphasis is reactive, but with a proactive element as time permits.

As long as the distributed LAN system is a "flat LAN", that is, all segments of the LAN are connected through local or remote bridges, the networking protocols and management protocols are the same as for the Classic LAN configuration.

The IBM LAN NetView family of products can be viewed as productivity tools for the LAN administrator. In the Massive LAN configuration, the number of LAN administrators and their scope of responsibilities control the deployment of the IBM LAN NetView products. Each LAN administrator would have a managing workstation at his or her disposal, with IBM LAN NetView Manage 1.0 installed. Additional applications would be installed based on the LAN administrator's duties. Controls exist within IBM LAN NetView Manage 1.0 to select which workstations are managed from a given managing workstation. The View component of IBM LAN NetView Manage 1.0 also provides a way to organize the information from the distributed LAN system along lines familiar to your organization.

A particular LAN administrator could focus on the LAN Servers or the Database Managers, while another administrator could handle Novell Servers (using IBM LAN NetView Manage 1.0 in combination with Novell's management application). Other LAN administrators could use IBM LAN NetView Monitor 1.0 to collect performance data and analyze it for trends. The results of this analysis could be used to predict capacity shortfalls before they occur, giving you time to address them.

A LAN administrator could be assigned to support a particular user department, managing all of the distributed LAN system resources used by that department. This administrator would use IBM LAN NetView Fix 1.0 to discover problems as they are occurring. Using the facilities of IBM LAN NetView Manage 1.0, the administrator would correct the problem and keep the users happy.

In this complex environment, the need exists to share data between managing workstations. This capability is enabled in the IBM LAN NetView family of products, but particular exploitations of this capability are left to you.

4.5 Branch Office

Often, a WAN is used to distribute information between multiple locations, where each location has its own distributed LAN system. Each branch location tends to have the same uses for its distributed LAN system as the other branch locations. The central site typically has one or more host processors to augment the branches' information needs. This Branch Office configuration is usually handled from a central site by a department devoted to LAN administration. Little, if any, LAN administration expertise is available in the branch office.

Where the CMOL management protocol could be used before, it can only be used within the branch offices in this configuration. Across the WAN, the IBM LAN NetView family of products supports either CMIP over TCP/IP (CMOT) or IPX (using Novell's management application). These may be used if the WAN can support TCP/IP or IPX in addition to the traffic currently being carried. Most WANs built using routers support using these protocols. However, if your WAN is based on SDLC, SNA management services must be used. Use of SNA implies using the S/390* management products, which are supported by the IBM LAN NetView family of products.

Presuming that your WAN can support CMOT or IPX management protocols, you can choose one or more approaches. If you are interested in managing only IPX resources in the branches, you might place one (or more) managing workstations at the central site. These stations would have IBM LAN NetView Manage plus the Novell managing application installed. If you wish to manage non-IPX resources, you could use the CMOT management protocol. Since the managing workstation can support multiple management protocols, you could use both IPX and CMOT if you had a combination of IPX and other resources to manage. Based on the number of resources to be managed, you may choose to focus only on critical resources such as servers or workstations running important line-of-business applications.

If your WAN is SDLC-based or if you wish to integrate your distributed LAN system management scheme with your enterprise-wide management scheme, you would place managing workstations in each branch. Here, the concept of the IBM LAN NetView family of products as personal productivity tools changes to a concept of "distributed management intelligence". The managing workstations would run mostly unattended, since no LAN administration skill is available in the branches. They would communicate with the instrumentation using CMOL or CMOT within the branch. Through the use of automation (using facilities in IBM LAN NetView Fix 1.0) and remotely-issued commands, the managing stations would attempt to keep the branch's distributed LAN system healthy. When problems occur that are beyond the automation capability implemented at the managing workstation, the managing function can issue a notification that makes its way to NetView on the mainframe via the IBM LAN NetView Tie product. The LAN administrator at the central site would use the NetView products to learn of problems. Using the remote operations capabilities of the IBM Communications Manager/2 product, the administrator would issue commands to correct the problem.

The distributed management intelligence concept is not limited to those configurations with SDLC WANs. Managing workstations can be assigned to particular scopes of responsibility, with automation provided that is specific to those responsibilities. Since managing workstations can also be managed, information from these distributed managing workstations can be collected and summarized at a central managing workstation. Commands may then be issued from the central managing workstation to the distributed managing workstations. Unfortunately, applications to perform collection and summarization do not exist, although the IBM LAN NetView family of products provides facilities to support such applications.

4.6 Remote Service

The Remote Service configuration is an interesting one. The implementer of the management scheme usually is an organization that sells servicing contracts to small, client organizations that do not care to manage their own distributed LAN systems. These contracts allow the client organizations to take advantage of distributed LAN systems without having to develop their own LAN administration skills. The service organization usually has several such contracts, forcing it to focus more on trend analysis and preventative maintenance than on immediate problems.

Access to the clients' distributed LAN systems is usually via a switched line, using The IBM LAN Distance* product or a similar product. As a condition of the service contract, the service organization installs instrumentation in the client's distributed LAN system. This instrumentation would include the IBM LAN NetView Enabler, IBM LAN NetView Agents Extended, IBM LAN NetView Agents for DOS, and other instrumentation products.

For managing function, the service organization can choose to follow the "personal productivity" model or the "distributed management intelligence" model. In the personal productivity model, IBM LAN NetView Manage 1.0 and applications such as IBM LAN NetView Monitor 1.0 would be installed at the service organization's site. At a regular interval, the managing workstation would dial into the client's distributed LAN system and collect data. The connection might be maintained for several minutes or hours, depending upon what kind of data is to be collected. In the event of a serious problem reported by a client, the service organization could dial into the client's distributed LAN system on demand to determine the situation and to take corrective action.

Using the "distributed management intelligence" model, the service organization would install unattended managing workstations in each client's distributed LAN system. These managing workstations would have IBM LAN NetView Manage 1.0, IBM LAN NetView Fix 1.0, and other applications (perhaps some specially written by the service organization). These applications would allow the managing workstations to handle trouble reports and gather data without intervention. Periodically, the service organization would check on the performance of these managing workstations to insure that they are meeting the client's needs. This checking could use the same polling technique described above. If the managing workstation detects a problem beyond its control, it could notify the service organization. Facilities in IBM LAN NetView Fix 1.0 allow a pager to be called if service personnel need to address a problem. A LAN administrator at the service organization's site can then react to the notification, using managing functions of the administrator's workstation or the managing functions of the managing workstation in the client's distributed LAN system.

A drawback of a Remote Service configuration is the speed of the switched line. Gathering large amounts of data results in long transmission times. However, use of the distributed management intelligence model allows the data to be summarized at the client distributed LAN system's managing station. Only the summary needs to be collected and analyzed. In this way the complete set of data would need to be retrieved only if a problem was encountered.

4.7 Summary

The distributed LAN system configuration types of Classic LAN, Massive LAN, Branch Office, and Remote Service are not meant to be exhaustive. Your particular situation will probably involve some combination of the approaches described for these configurations. The richness of the IBM LAN NetView family of products provides you ample flexibility so that you can implement a management scheme appropriate to your distributed LAN system's configuration.

Chapter 5. LAN NetView Hardware Requirements and Performance

This paper describes the hardware requirements in terms of memory and disk space needed to use the LAN NetView family of products, and discusses the performance aspect using various configurations and example scenarios. This type of information is vital to marketing people who are responsible for providing systems management solutions in a client/server environment. One of the major factors in determining the right configuration for an installation is defining what the minimum requirements are, and how to attain maximum performance. In this capacity, the paper provides the guidelines for making this exercise as simple as possible when the LAN NetView family of products is chosen as the proposed solution.

5.1 Introduction

The fixed disk requirements and random access memory requirements for the LAN NetView products depend on whether the system involved is a managed node or a managing node (or both), and on the applications and operating subsystems installed. Worksheets for memory and disk requirements are given here and are used to determine the hardware requirements for several example scenarios.

For a managing node (LAN NetView Manage Version 1.0 and LAN NetView Applications), memory required is estimated at 3-13 MB above any prerequisite or co-requisite software, depending on the LAN NetView applications installed and concurrently used.

The recommended minimum hardware requirements for an OS/2 managing node including any prerequisite or co-requisite software is a 486SX (or equivalent) with 16 MB of memory. Additional memory or a faster processor may provide better performance, especially if the system is not dedicated to the LAN NetView product's management. The minimum recommended disk space is 300 MB.

For an OS/2 managed node (LAN NetView Enabler Version 1.0 and LAN NetView Agents Extended Version 1.0), the memory requirements depend on whether the agents are idle (waiting for management direction) or busy processing a request. Since the agents are idle except when a specific request comes in, or when emitting an event, OS/2 managed system memory requirements are shown for the idle state in the scenarios below.

1 MB of additional memory above other prerequisite or co-requisite software is recommended for a managed OS/2 workstation. For systems which may have frequent monitoring or management requests, for example servers, 2 to 3 MB of additional memory above prerequisite or co-requisite software is recommended. Management requests occasionally can increase LAN NetView Enabler 1.0 memory use to 4 MB depending on the number and type of agents and frequency of their use. Additional memory for an OS/2 managed system may produce better system performance.

The recommended minimum memory requirements for a DOS managed node including any prerequisite or co-requisite software is 1 MB. Memory requirements for the DOS agent depends upon the availability of Expanded

Memory Specification (EMS) memory or a DOS Extender product such as Quarterdeck's** QEMM**, Qualitas** 386MAX**, or Qualitas BlueMAX**.

The recommended minimum memory requirements for a DOS Windows managed node including any prerequisite or co-requisite software is 3 MB.

5.2 Basic Disk and Memory Requirements for LAN NetView Manage

The following section is taken from the README file for LAN NetView Manage 1.0:

Determining Disk and Memory Requirements. The LAN NetView hardware requirements are additional above other prerequisite or co-requisite software. To determine the LAN NetView disk requirements, sum the disk requirements of each installed component:

	Disk Usage (MB) LAN NetView Manage	LAN NetView Enabler
Services (base):	11.1	6.9
Communications protocol	0.9	0.5
Services (optional):		
Toolkit	8.5	
Discovery communications:		
CMOT/SNMP	0.1	
NetWare	2.1	
View component	2.7	
Agents:		
OS/2	0.3	0.3
LAN Requester	0.4	0.4
Agents Extended:		
Communications Manager	0.3	0.3
Database Manager	0.2	0.2
LAN Server	0.4	0.4
Applications:		
LAN NetView Fix	2.5	
LAN NetView Monitor	3.5	
LAN NetView Tie	1.2	
Runtime disk usage	15 Min.	5 Min.

Note: The runtime disk usage varies widely with applications and functions used.

To determine the LAN NetView products' overall memory requirement, sum the memory requirement of each component that is concurrently used.

For a managed node, use the numbers in the Idle column as the guideline for memory requirements. When a node with LAN NetView Enabler is busy (that is, it is answering a management request or emitting a notification), as much as 2MB over the idle requirements are temporarily used.

For a managing node, use the numbers in the Busy column as the guideline for memory requirements. For a LAN NetView Manage 1.0 node, the Idle numbers are included in the Busy numbers. For a LAN NetView Manage 1.0 node that is also managed, add the numbers in the LAN NetView Enabler Idle column from the Agents section.

	Memory Usage (MB)		
	LAN NetView Manage		LAN NetView Enabler
	Idle	Busy	Idle
Services (base):		2.0	
Postmaster	0.4		0.3
Event sieve agent	0.2		0.2
System agent	0.1		0.1
Services (optional):			
Master and slave mode of object registration service	0.2		0.2
Discovery and topology	0.3	1.3 (Note 2 and 3)	
View Component	0.9	1.9 (Note 1)	
Agents and Agents Extended:			
OS/2			0.1
LAN Server or LAN Requester			0.5
Database Manager			0.4
Communications Manager			0.3
Performance (choose one):			
Idle			0.1
Log to local file			0.4
Graphing on Manage node in real time			1.6
Applications:			
LAN NetView Monitor	0.1	2.6	
LAN NetView Fix	0.1	3.0	
LAN NetView Tie	0.4	1.0	

Notes:

1. Add 5KB for each discovered system in the View component.
2. Add 3KB for each discovered system.
3. Environments that include simultaneous IPL or restart of many systems with LAN NetView Enabler can use more memory than shown to improve the performance of discovery.

For more information about the hardware requirements of prerequisite and co-requisite software, see:

- *OS/2 2.X Getting Started Manual*
- *OS/2 Extended Services 1.0 Information and Planning Guide*
- *LAN Server 3.0 Network Administrator's Reference*
- *Communications Manager/2 Information and Planning Guide*
- *OS/2 TCP/IP Installation and Maintenance*

5.3 Scenarios

The following are some typical scenarios showing the disk space and memory needed.

5.3.1 Case 1: OS/2 Managed Server System

In this case, the managed system is a LAN server or Database server. It has management agents for OS/2 and either LAN Server or Database Manager. Additionally, the server performance is important and the performance agent is used. The most common performance monitoring function is logging performance data to be retrieved by the managing node at times when the server has light usage. Occasionally, the system administrator may want to view the server's performance in real time from the managing system (via LAN NetView Monitor real time graph).

For the LAN NetView product to have the minimum performance impact over present server performance, choose the higher memory estimate which includes real time graphing. However, if real time monitoring is expected to be an infrequent occurrence and the server has enough memory to accommodate the additional 1.2 MB used during real time graphing, then choose the lower memory estimate as the hardware requirement.

Table 2. Disk Requirements - OS/2 managed system with LAN Server or Database Server

LAN NetView elements	**LAN Server**	**Database Server**
LAN NetView Enabler	6.9	6.9
Communications Protocol	.5	.5
OS/2 agent	.3	.3
LAN Server agent	.4	-
Database Manager agent	-	.2
Total for LAN NetView static elements	**8.1**	**7.9**
Runtime disk space (varies with usage)	**5 (minimum)**	**5 (minimum)**
Total LAN NetView System disk requirements	**13.1 MB**	**12.9 MB**

Table 3. Memory Requirements - OS/2 managed system with LAN Server or Database Server

LAN NetView elements	**LAN Server (idle)**		**Database Server (idle)**	
Services (base)	.6		.6	
Object Registration Svc (optional)		(.2)		(.2)
OS/2 agent	.1		.1	
LAN Server agent	.5		-	
Database Manager agent	-		.4	
Performance agent (logging to local file)	.4		.4	
Delta for realtime graphing (busy state)		(1.2)		(1.2)
Total for LAN NetView required elements	**1.6 MB**	(3.0)	**1.5 MB**	(2.9)

Note: if the servers are expected to be observed real time using LAN NetView Monitor, add 1.2 MB for realtime graphing.

5.3.2 Case 2: OS/2 Managed Workstation with LAN Requester or Communications Manager

In this scenario, the managed system is a user workstation with LAN Requester or Communications Manager. The management agents for each installed management object are included (OS/2 and either LAN Server or Communications Manager).

The performance agent is started but is not typically collecting performance data. When a performance problem arises or when the system administrator has a particular interest in this workstation's performance then performance data will be collected and logged. The memory requirement for this workstation is then typically about 1.3 or 1.1 MB (that is, no performance data being collected or logged). For minimum performance impact of the LAN NetView product to the workstation even when logging performance data, 1.6 or 1.4 MB of memory should be provided.

When this workstation answers a management request received from the managing node, or emits a notification, as much as 2 MB of additional memory will be used during the event. These events are expected to be infrequent and of short duration, and the short-term use of this memory is not expected to cause system performance problems. However, if a workstation has marginal memory for its present usage, the occasional management events may be a noticeable disruption and the additional 2 MB may improve normal operation as well as accommodate management events.

Table 4. Disk Requirements- OS/2 managed system with LAN Requester or Communications Manager

LAN NetView elements	**LAN Requester**	**Comm. Mgr.**
LAN NetView Enabler	6.9	6.9
Communications Protocol	.5	.5
OS/2 agent	.3	.3
LAN requester agent	.4	-
Communications Manager agent	-	.3
Total for LAN NetView static elements	**8.1**	**8.0**
Runtime disk space (varies with usage)	**5 (minimum)**	**5 (minimum)**
Total LAN NetView System disk requirements	**13.1 MB**	**13.0 MB**

Table 5. Memory Requirements - OS/2 managed system with LAN Requester or Communications Manager

LAN NetView elements	**LAN Requester**		**Comm. Mgr.**	
Services (base)	.6		.6	
OS/2 agent	.1		.1	
LAN Requester agent	.5		-	
Communications Mgr agent	-		.3	
Performance agent	.1		.1	
logging to local file		(.3)		(.3)
Total for LAN NetView required elements (idle)	**1.3 MB**	(1.6)	**1.1 MB**	(1.4)

Memory usage for performance agent logging to file is 0.4 MB. The table included 0.1 for idle, so the additional for logging to file is 0.3.

5.3.3 Case 3: OS/2 Managing System with Communications Manager and LAN Requester

This case estimates a managing system with LAN NetView Manage 1.0 and applications pilot installation used by a power user whose purpose is to learn the LAN NetView products′ capabilities and how to best apply them for the business′ needs. A system dedicated to LAN NetView Manage 1.0 for system management is recommended and a fast 486 processor will provide improved responsiveness over the minimum recommendation of a 486SX.

For memory requirements, this scenario environment includes a single LAN management protocol and provides for concurrent use of multiple management applications. The management system is expected to manage itself and therefore management agents are included. Memory space for 125 systems is included.

If for example, the prerequisite and co-requisite product memory requirements total 10 MB, then a total 24 MB of memory is needed for this pilot installation managing machine .

For disk requirements, a generous amount of space is suggested for storage of experimental performance and fault data.

After the expected business usage of a LAN NetView Manage 1.0 managing system is determined, lower hardware requirements may be identified.

Table 6. Disk Requirements - Managing system w/Comm. Mgr.+ LAN Requester, all managing applications

LAN NetView elements	**Disk**
Services	11.1
Communications Protocol	.9
LAN NetView Fix application	2.5
LAN NetView Monitor application	3.5
LAN NetView TIE application	1.2
OS/2 agent	.3
LAN Requester agent	.4
Communications Manager agent	.3
Database Manager agent	.2
CMOT Discovery (Note 2)	.1
View component	2.7
Total for LAN NetView static elements	**23.2**
Runtime Disk space (varies with usage)	**15.0 (minimum)**
Space for Experimental Data	**100.0**
Total LAN NetView System disk requirements	**138.2 MB**

Notes: Disk space for the Toolkit on a developer′s machine is 8.5 MB

Table 7. Memory Requirements - managing system with Communications Manager and LAN Requester	
LAN NetView elements	**Memory (MB processing)**
LAN NetView Manage (includes single communications protocol)	2.0
View component (note 1)	1.9
Discovery/topology (notes 2-3)	1.3
Discovery/View data for 125 systems (notes 1,2)	1.0
OS/2 agent	.1
LAN requester agent	.5
Database Manager agent	.4
Communications Manager agent	.3
Performance agent	.1
Managing Applications	
LAN NetView Monitor	2.6
LAN NetView Fix	3.0
LAN NetView Tie	1.0
Total for LAN NetView elements	**14.2 MB**

Notes:

1. Plus 5K bytes per discovered system in View
2. Plus 3K bytes per discovered system
3. Environments which include simultaneous IPL or restart of many systems with LAN NetView Enabler can use more memory than shown to improve the performance of discovery
4. This worksheet shows both LAN NetView Monitor and LAN NetView Fix being used concurrently. If they are used serially the memory requirement can be reduced.

5.3.4 Case 4: DOS Version 5.0 Managed System with LAN Support Program and DOS LAN Requester:

LAN NetView has two DOS components, the agent and HLM. Their combined memory is 85KB with typical configuration parameter settings. However, since all of the 85 KB is enabled for placement in extended memory, a memory manager may be able to fit much of the programs into extended memory resulting in little affect on the DOS application space.

LAN Support Program is a pre-requisite product for LAN NetView and together with DOS LAN Requester require about 120KB of memory. However, since all of the 120 KB is enabled for placement in extended memory, a memory manager may be able to fit much of the programs into extended memory.

With LAN NetView DOS agent, HLM, LAN Support Program and DOS LAN Requester in a typical OS/2 machine with XGA video and a token-ring card, the BlueMAX memory extender was able to locate 152 KB in extended memory resulting in a DOS application space reduction of 53 KB. The results obtained will depend on the installed hardware, its configuration and the memory manager used.

The disk space requirement for the LAN NetView DOS agent and HLM is 250 KB.

Use of well known DOS performance improvements like a disk cache and/or a RAM disk will also improve the performance of LAN NetView Agents for DOS.

5.4 Disk Requirements at Runtime

In addition to the installed code, there are additional fixed disk requirements for databases established at runtime. This depends on the managed client activity, and was estimated for the above scenarios. The significant contributors to this fixed disk requirement include swap file space, the LAN NetView Monitor database and log files, and Fix log files.

5.5 Performance

The following sample performance figures are for a 33 or 50 mhz 486 managing node, and a 20 mhz 386 managed node.

```
                                          Response time (seconds)
25 mhz:
 LAN NetView TIE process 100 Alerts:               10.8

33 mhz:
 Simple get of single attribute                      .2
   (measured at the XMP API)
 Scoped get of all (40) LS attributes              3
   (measured at the Request Mgr display)
 Scoped get of all (5558) OS/2 attributes          40
   (measured at the Request Mgr display)

50 mhz:
 LAN NetView FIX display a client event            2.4
 LAN NetView FIX log a client event                3.3
```

The response time for the scoped get at the API level is representative of the time an application will see for a scoped get of a single attribute. The scoped get of all attributes of the LS and OS/2 agents illustrates two points. First, getting all attributes can result in a large number of attributes and a noticeable amount of time. Second, the time per attribute usually decreases with the number of attributes retrieved increases.

5.5.1 Performance Tips

The time required for the View component to display all machines can be greatly improved by operator awareness and action in one special situation: in an environment where many managed machines are started nearly simultaneously (as in a test lab), let the managed machines finish their IPL and initialization so they may be known to (that is, discovered by) the managing machine before opening the VIEW folder.

For the LAN NetView FIX application, the time to display an event on the console is affected by the maximum number of events that can be displayed on the console. This number is set in the file \ibmfix\fxconfig.dat by the parameter

MaxEvents. The default value is 20, the minimum is 3, and the maximum is 99. The smaller the number, the less time taken to display an event.

It is also possible to improve performance for certain operations significantly by positioning the directory name used for the LAN NetView product at the beginning of the LIBPATH statement in CONFIG.SYS.

5.6 CPU Utilization

The following are measured utilization for a 50 mhz 486 managing node and managed node:

```
For a managed node:
    Idle utilization:                                          < 1%
    Utilization when LAN NetView Performance agent
      logs to a local file:                                    < 5%
    Utilization when LAN NetView Performance agent
      sends data to Monitor real time graph:                   < 8%

For a managing node:
    During management activity such as, discovery,
      View displaying management collections or
      LAN NetView Fix registering sieves:            100% (CPU bound)
```

5.7 Capacity

The LAN NetView product has been tested with over 100 workstations. The maximum number of workstations that can be effectively managed by one managing node in practical environments depends on work load characteristics and the combination of LAN NetView applications and features being used.

Rates of management events from managed workstations will vary depending on the activity of the workstation.

5.8 Summary

The memory and disk space requirements for a LAN NetView managing machine vary greatly, depending on the functions performed. Upgrading from the minimum recommended managing machine (486SX, 16 MB of memory) will result in improved performance.

The memory and disk space requirements for a LAN NetView managed machine vary, depending on the functions performed. Upgrading managed servers from the minimum recommended, 1 MB additional memory may improve server performance with LAN NetView.

Performance statistics and some hints on improving performance have also been included.

Chapter 6. Hints and Tips for Using Topology Management Services

6.1 Introduction

This paper describes how to set up and run the LAN Netview product's discoveries and Topology Management Services (TMS) to maximize the discovery and storage process. This paper presents hints and tips on how to achieve optimum discovery capability with TMS. The result should be a faster, more efficient managing workstation running TMS.

Also discussed will be the optimal use of discoveries and the LAN NetView View user interface component as they pertain to TMS. Hints on how to sequence the startup of these features in order to optimize the network and the managing workstation are presented.

6.2 Scope

This paper is directed to administrators or people who perform at least one of the following functions:

- Set up managing networks
- Set up the managing node of a managed network
- Responsible for running the TMS environment.

Prerequisite knowledge of the LAN NetView Manage and Enabler products is assumed, since the intent here is to provide helpful information which relies on a basic understanding of the LAN NetView platform functions.

6.3 Topology Management Services

The LAN NetView Manage product's Topology Management Services (TMS) is a service that accumulates information about discovered nodes in its surrounding environment. The phrase "surrounding environment" is used instead of "network" because TMS literally knows no boundaries. If a node is found by any of the discoveries, LNTCPIP (TCPIP Discovery), LNCMOL (CMOL discovery), or LNNETW (NetWare Discovery), it could be within the same local network, across a bridge, or 8 hops away. TMS simply accepts the information about the surrounding node and places it in the database. Once this information is stored in TMS, any application can, using the XMP (X/Open Management Protocols) API, send queries to the TMS database for this information. One such application is the LAN NetView Manage View component. The View component queries TMS for all of the nodes TMS has and all of the information under each node. It then displays the nodes in icon format.

To optimize discoveries and the managing node running TMS, the discoveries must first be limited in their scope. You must first ask the question, "What do I want to manage from a TMS standpoint?". To elaborate, is it necessary for you to run a discovery, find over 100 nodes, gather information about these nodes, flood the network with requests for information about these 100 nodes, only to have the real intention of managing only 10 nodes? You have just created the overhead of approximately 90% extra processing for nothing. And the waste does not stop there. The LAN NetView View component, when executed, queries

TMS for all physical systems (100 in this example), and then calls back for all information for each system, resulting in 101 total queries! Also, TMS is tracking events from each of these nodes and the View component is also tracking these events by receiving changes from TMS.

Thus the goal should be to discover only those nodes you actually wish to manage. This paper will not attempt to delve into the assorted masking techniques of the discoveries. Instructions of use and examples can be found in the *LAN NetView Administration Guide.* It behooves the user to read about these masks and also the use of the ACCESS file. This file keeps any extraneous managing node from discovering nodes in your network and possibly adding to the managed node's workload. Using these masks in conjunction with the ACCESS file will help you shape your network or cell to a well-run and efficient environment.

6.4 Setup of Managed Nodes

It is imperative that all OS/2 and/or DOS managed nodes are up and running and that the LAN Netview System Agent (LNSYSAGT) on the OS/2 nodes is running before TMS is running on the managing node. The reason for this is that the LNSYSAGT must be running in order for TMS to perform a get request to it for the distinguished name of that node. If the LNSYSAGT is not up, TMS does not know the node is running LAN NetView and does not retry. Also, TMS cannot place a remote sieve on that node without the distinguished name. Without this remote sieve, TMS cannot listen for any future object creation events.

The LNSYSAGT must be brought up first before any of the LAN NetView agents because it listens for object creation events. If an agent is brought up first and generates an object creation, the LNSYSAGT, when brought up afterwards, will not hear the event. Therefore, the LNSYSAGT will not be aware of the agent's existence. Also TMS had not registered an event sieve yet, so TMS will not hear the object creation event, either.

Once TMS has performed a successful get request to the LNSYSAGT of that node, TMS will then place a remote sieve on that node. If an object creation event occurs, as possibly a result of one of the LAN NetView agents coming on-line, TMS will perform a get request for that new object.

It is also useful to shut your managed node off to other managing nodes and only answer to your managing node. This will cut down the activity generated by your managed node having to answer the requests from other managing nodes as well as your own. For example, if your managed node has been discovered by two managing nodes, it will receive a get request from both managers when an object creation event is generated because you brought up another LAN NetView agent. You must understand that the managed node can become severely hampered by the amount of extraneous requests if the number of discovery managers requesting information grows. The ACCESS file effectively shuts out all other managing nodes except the node from which management is desired.

6.5 Ordering Startups

A most important point if you are planning to run CMOL discovery is that you should run it first for a while until discovery activity dies down, before you run TMS. CMOL discovery broadcasts a message to the network and receives a reply from each CMOL node that hears the broadcast and responds successfully. As CMOL discovery receives each response, it collects them in its own database. Performing this broadcast causes some intense network traffic and therefore should be done with minimal services running simultaneously. Keep in mind that CMOL discovery needs to only be run once, or until you have accumulated whatever nodes you need in the CMOL database.

To see what nodes have been accumulated, use the CMOLUTIL utility. A description of the utility and instructions of it's usage can be found in the *LAN NetView Administration Guide.* Also, this is a good time to remind you of the masking available for CMOL discovery. You can also use the CMOLUTIL utility to filter out unwanted nodes and populate the CMOL database with only those nodes you want TMS to discover.

Note: Perform deletes or garbage collection on the CMOL database while LAN NetView Manage is down, or after TMS has had time to read the file. This action will lessen the contention factor between TMS reading the CMOL database and any shrinking or changes.

In order to regulate the startup of CMOL discovery and TMS, you can wait to start TMS manually instead of using the startup file for SVSTART. You can specify not to start the application in the LRF file before you perform an SVADDOBJ with the LRF. Once you change LNTOP.LRF for NO_START and perform an SVADDOBJ, you can run SVSTART to initialize and start the framework and CMOL discovery. As mentioned earlier, once you have whatever nodes you need in the CMOL discovery database, you can stop CMOL discovery. You can then start TMS by issuing the command **SVSTART topology**. TMS will read the CMOL discovery database upon initialization and perform a get request to the node. If the node responds successfully to the request, the node will be placed in the TMS database. If the node is not responding for any reason, it will not be put in the TMS database. This action conforms to the other discoveries which only send TMS active node information.

Since CMOL discovery is the only passive discovery (TMS must retrieve node information initially), TMS must filter out all inactive nodes. Therefore, there might be a difference in the number of nodes in the CMOL database versus the TMS database. TMS only reflects active nodes, whereas CMOL retains all nodes it has found, and nodes entered by the user with the CMOLUTIL utility. If a CMOL node is not placed in the TMS database, re-IPL'ing the node will result in the node sending another automatic response which CMOL will ignore, but TMS will act upon.

Note: The other two discoveries, LNTCPIP and LNNETW must be run concurrently with TMS. They communicate the discovered node information to TMS directly.

Finally, the LAN NetView Manage product's View component queries the TMS database for all systems TMS has discovered. It is highly recommended that you do not start the View component until TMS has slowed down in the discovery of nodes. This will enable the discovery process and the storing of nodes in the TMS database to flow smoothly with minimal network traffic.

Starting the View component after topology has run for awhile will minimize the processing impact it has on topology as well as the framework.

6.6 Distributed Discoveries

Since all discoveries vie for processing and have extended memory requirements, it would probably be more fruitful to run the discoveries on a separate machine so TMS can run in a minimum environment. Even though it is an unannounced and unsupported feature, the discoveries, LNTCPIP and LNNETW communicate with TMS via the XMP API. This API allows these applications to communicate in a distributed environment. You can run these discoveries on a separate machine provided that you add the LRF entry for topology by using another node's host name that is running topology.

For example, I have nodes X and Y. I want node X to run topology, but I want to run TCPIP discovery on node Y. To do this, I would use the SVADDOBJ command to add the LNTOP.LRF to the ORS (Object Registration Service) on node X using the default host name. On node X, here is what I would type: **SVADDOBJ LNTOP.LRF**. On node Y, I would simply add LNTOP.LRF the same way, but I would specify a target host name because it is not the default host name (in this case, Y). Therefore, I would type: *SVADDOBJ LNTOP.LRF X*. Node X is now the node that the ORS on node Y would direct all topology requests or actions. The *LAN NetView Manage Administration Guide* has an explanation of how to register an object in the ORS with a target host name other than your local node. The information is listed under the SVADDOBJ command in the Command Reference chapter. With this ability, you can now dedicate one machine to topology and use other machines for running the discoveries. This will result in a faster executing environment for both topology and the discoveries.

Because TMS reads the CMOL database information directly, CMOL discovery and TMS must remain on the same machine. Thus CMOL discovery should be run first, however, it can also be run concurrent with TMS. The results of running concurrently will probably be a slower environment. The optimum use is to run CMOL discovery until discovery settles to a minimum, and then while CMOL discovery is still running, start TMS. This can be achieved by simply performing an SVSTART on TMS.

6.7 Distributed View

As with the two discoveries, the LAN NetView View component also communicates with topology by means of XMP. XMP allows the View component the same luxury of running on another node while aiming at the remote machine running TMS. Because of this distributed capability, you can run multiple Views if you have a number of machines, each with one View pointing to the topology machine.

However, a caveat exists as to running more than one View. Remember the earlier scenario with 100 nodes? Now you must multiply the number of queries by the number of Views querying TMS. For example, five machines running View pointed at a topology machine with 100 nodes in it's database will result in 505 initial queries. Therefore to minimize the load on TMS, cut down the amount of nodes needed to be stored in topology. Also, it is imperative to cut down on the initial traffic. To achieve this goal, bring up the Views serially. In other words, bring up the first View, allowing it to fully seed that View before starting the next View. This may take a while longer, but considering that you will perform this action only occasionally, it seems inconsequential.

Note: Once again, the distributing of the View component and the discoveries is an unannounced and unsupported feature. It is offered as a helpful tool to use.

6.8 Summary

It is hoped that this paper gives you some hints and tips into how to perform a successful discovery process with minimal impact to your network. As with any discovery, initialization is sometimes the most active period in the execution of that software. This paper's intent is to show what you can do to help minimize any possible "bottlenecks" so that full discovery is achieved in as little time as possible.

Chapter 7. Comparison of LAN NetView Monitor 1.0 and SPM/2 2.0

This paper explains the benefits of using the Monitor product on the LAN NetView framework versus using the System Performance Monitor/2 V2.0 standalone product for LAN Performance management. The paper is aimed at current users of the SPM/2 2.0 product, and other competitive products like SPM/2.

7.1 Introduction

With the introduction of the LAN NetView family of products, there are now two IBM products that provide the user with OS/2 2.x-related performance information: LAN NetView Monitor Version 1.0 and System Performance Monitor/2 Version 2.0 (SPM/2 2.0). This raises the following questions: Which product should a customer use and why? What is the benefit of one product over the other?

The first step in answering these questions is to understand the intended audience for each product. LAN NetView Monitor product is intended for use by system administrators, while SPM/2 2.0 is primarily targeted for use by developers and analysts. The needs of these audiences differ as described below:

- Systems administrators are responsible for managing existing, production environment LANs. Their task is to ensure that the LAN runs well — that performance objectives are met and capacity objectives are not exceeded. Since there are many systems to be watched, the administrator does not want to be continually flooded with performance data. Instead, they are primarily interested in information that would alert them to a potential problem. Detailed performance information is still needed, but only when specifically requested by the administrator for the purposes of problem determination. Summarized information is also needed to provide help in observing trends in resource usage over time.

  Hence capabilities that are important to the administrator include centralized management, threshold monitoring, forwarding of alerts to a central location, graphing facilities that provide a "quick look" into a system's current performance, and reporting functions that allow a large variety of reports to be generated (for purposes that range from problem isolation and analysis, to trend and capacity analysis).

- Application developers and performance analysts, on the other hand, generally work in either a standalone environment, or a relatively small, well-controlled LAN environment. Their goal may be to tune an application under development or to solve a very specific performance problem in a relatively small environment. Their primary need is for *information* — detailed system information down to the process and thread level. Tools are also needed to view and analyze this information — tools such as graphing and reporting utilities. Since these people are closely watching the systems being monitored, there is no real need for capabilities such as threshold monitoring and forwarding of alerts.

The following sections will describe the functions and features of SPM/2 2.0 and LAN NetView Monitor Version 1.0 within the context of their respective target audiences. SPM/2 2.0 will be addressed first since LAN NetView Monitor Version 1.0 provides additional function over that provided by SPM/2 2.0.

7.2 SPM/2 2.0: Benefits for the Developer/Analyst

SPM/2 2.0 is a relatively small tool that aids the developer and performance analyst by collecting data for analysis. It supports systems with OS/2 2.0 plus the first Service Pak (level XR06055) and beyond, or any version of OS/2 2.x beyond this. Utilization information on the following base operating system resources is supported:

- CPU (to the process/thread level)
- Physical disk
- RAM, including Page In/Out activity (page faults to the process/thread level)
- Files (on a per process/thread basis for each file)
- HPFS and FAT Cache
- Printer port
- Communications port

LAN Server 3.0 and LAN Requester 3.0 data can also be collected by SPM/2 2.0.

Data is logged to a file from which a set of standard reports can be generated. A subset of the resources (CPU, Disk, RAM, Paging) can also be viewed at a system level via the graphing utility — either realtime, or historically from log files. If the resource of concern is RAM, the memory analyzer, Theseus2*, can be used to obtain even more detailed memory information. This memory analysis tool is particularly interesting to programmers and analysts, since information is provided down to the OS/2 control block level.

From the graphs, reports, and information produced by the Theseus2 memory analyzer, the user can narrow in on the specific applications, processes, and resources that may be causing the performance problem. Given this information and a knowledge of the code involved, solutions can be proposed.

The following functions are provided only in SPM/2 2.0 and are not supported in the LAN NetView Monitor product due to the fact that they would only be of interest to a developer or an analyst:

- Theseus2 Memory Analyzer

  This standalone OS/2 Presentation Manager tool provides application developers and analysts with in-depth insight into OS/2 memory management. An important feature of the Theseus2 product is the capability to provide working set information on a per-process basis. (Working set is the measure of the number of memory pages that have been accessed over a specific time interval.) Through the use of the Hyperblock* product for linking, the user can navigate from one memory control block or memory location to another by double-clicking (with the mouse) on the highlighted address of the location to be viewed. Memory locations containing executable code can be dis-assembled for debug. An upgrade to the Theseus2 memory analyzer (available on OS2BBS, and the CompuServe** information service) also includes a memory leak detector, which is very valuable to a developer.

- Application Programming Interfaces (APIs)

  - A user may need to pull data directly into a line of business application for manipulation by that application, rather than use SPM/2 2.0's recording, graphing and reporting facilities for accessing the data. SPM/2 2.0 provides an API to the realtime collected data so that this can be accomplished.

– Another API enables a developer to define performance metrics unique to their application, and update those metrics from their application. Once defined and registered, SPM/2 2.0 can collect these application-specific metrics for user analysis.

7.3 LAN NetView Monitor: Benefits for the Administrator

With the exception of the developer-specific functions in SPM/2 2.0 (that is, the Theseus2 memory analyzer and the API's), LAN NetView Monitor 1.0 contains all the capability of SPM/2 2.0 and more. Each product contains the basic functions of data collection, graphing and recording. Beyond that, LAN NetView Monitor 1.0 has much additional function that is oriented toward the systems administrator environment. This additional function is described in the following sections.

7.3.1 Automated Operation

In order to understand how the LAN NetView Monitor product automates various aspects of performance management, the concept of the "policy" must be understood. LAN NetView Monitor 1.0 performance management is controlled through the use of policies, and a policy is simply a specification of all the characteristics of a monitoring session, including the following:

- Resources to be collected
- Threshold settings and actions
- Data collection schedule
- Data collection interval (once data collection is started)
- Whether or not to log data
- Log file transfer time
- Database storage options
- Database maintenance options

Unlike SPM/2 2.0's "Monitor Session Description" (which is similar to a "policy"), the nodes to be monitored by the LAN NetView Monitor product are not specified as part of the policy itself. Rather, groups of nodes to be managed are defined through the LAN NetView View interface, and are called Management Collections. Associated with each Management Collection is a performance policy, which the user customizes as they desire.

When a policy is started, an instance of the performance agent (called "Monitor/Scanner"), is started at each node to be managed. It is this agent that provides much of the LAN NetView Monitor product's automatic operation. Each instance of Monitor/Scanner has a copy of all the settings in the policy, and is responsible for starting and stopping data collection according to the schedule that was specified in the policy. When it's time for data collection to begin, Monitor/Scanner communicates to the underlying data collection code exactly which metrics should be collected, as well as whether or not the collected data should be logged to a file on the managed system. While data is being collected, Monitor/Scanner also checks for thresholds being exceeded, if any were defined in the policy. If a threshold is exceeded, Monitor/Scanner sends an alarm back to the managing system.

There is a side benefit of data collection being automatically controlled at the managed system. With this design, it isn't necessary for the managing system to be "connected" to the managed system all the time. Unless thresholds are being monitored, the only time connection is required is at the log file transfer time. Also, since data is being collected at the managed system, traffic back to the managing system is minimized (unlike SPM/2 2.0 where all the data flows back to the managing system), and thus this product is more suited to a large network environment.

Another aspect of automatic operation is the automatic log file transfer, which includes automatic database maintenance. This process is controlled from the managing system by a component called "Collection Control." Collection Control keeps track of the log file transfer times for each active policy, and initiates the transfer of these files to the database when the time occurs. As the data is transferred, it is stored in the database. The data is either stored in raw form (at the interval it was collected), or it can be summarized at an interval specified in the policy. Also as part of the database storage operation, existing data is summarized into daily, weekly and monthly summary records. The underlying detailed data is automatically deleted according to retention periods specified in the policy.

The last aspect of automatic operation is in the area of recovery, when either the managing or managed system goes down during operation.

- If a managing system goes down, when it is re-started, Monitor will search the database (which contains operational data as well as performance data) to see which policies were previously started. It will then re-issue "start" commands for those policies to ensure that they're still started.
- If a managed system goes down, the next time it powers up, a message will be sent back to the managing system indicating that it is back on-line. Monitor on the managing system will find all the active policies that include that node and re-start those policies on that node.

7.3.2 Threshold Monitoring

Threshold monitoring allows an administrator to "manage by exception," so that they don't have to closely watch every system being monitored. By setting up thresholds, the administrator can ignore managed systems until an event of interest occurs. Additionally, the user can reduce the overhead on the managed system by *only* monitoring thresholds — that is, they can leave logging "off" until something of concern occurs. At the point that a threshold is exceeded, a graph could then be started, or another policy initiated to start up logging.

In the Monitor policy definition, the user can specify a threshold value and severity for any metric included in the policy. As previously described, during performance data collection at each managed system, the "Monitor/Scanner" agent compares realtime data values against the defined threshold levels. When a threshold is exceeded, an alarm is forwarded back to the managing system where one or more of the following actions is taken, depending on the policy definition:

- A message can be displayed in an "Attention Window."
- An alarm can be stored in the performance database so that an alarm report can be generated at a later time.

- A user-specified program can be executed at the managing system. An example might be a pager routine that would phone the system administrator when a threshold is exceeded.

Critical threshold alarms can also be routed to NetView on the host by registering the threshold alarms of a specific policy and node with LAN NetView Tie 1.0. Once registered, the critical threshold alarms from that policy and node will flow to LAN NetView Tie 1.0 (as well as back to the managing system), and LAN NetView Tie will convert the alarms to SNA alerts and route them to NetView.

7.3.3 Database Storage

Storing performance data in a database is important to an administrator for several reasons. First of all, it provides a central repository for all systems being monitored, making it easy to generate an historical record of activity over time for a large number of systems. This information is useful for analysis of performance trends and problems, as well as load balancing and capacity planning tasks.

Secondly, the SQL interface to the data enables an administrator to customize their reports. This allows the administrator to produce reports on any subset or combination of data they're interested in, in addition to the standard reports provided with the Monitor product. Data can also be exported from the database, and then imported into a spreadsheet program for further analysis.

Because of the large amount of data that can be collected and stored in the database, old performance data will need to be periodically deleted. Hence, a database maintenance function is an important feature to include in the product. Due to the SQL interface to the database, this function is relatively easy to provide. LAN NetView Monitor 1.0 provides both automatic and manual database maintenance facilities:

- At logfile transfer time, the automatic maintenance function summarizes existing data into daily, weekly, and monthly summary records, and deletes detailed data that is older than the retention period specified in the policy.
- A manual utility is also provided that allows the user to delete any performance or alarm data from the database that they choose. This OS/2 Presentation Manager-based utility lets the user select the policy, node(s), and date(s) of interest, as well as the type of data to be deleted (detailed, summary, daily, weekly, or monthly).

7.3.4 Integration into the LAN NetView Platform

The LAN NetView Manage and LAN NetView Enabler products provide a standards-based platform for the creation and implementation of systems management applications for the LAN workgroup, and includes basic management function which complements any application written to the LAN NetView platform. The major components of this management platform are:

- LAN NetView View graphical user interface
- Management Application Programming Interface
- Topology / Discovery functions
- Remote Command Line facility
- Event Management Services
- Request Manager (utility to manipulate MIBs)

As an application written to this platform, LAN NetView Monitor 1.0 gains many advantages over "single-tool" type products (such as SPM/2 2.0), that must provide all of the function within the product itself. Following are benefits which the LAN NetView Monitor product gains from being integrated into the LAN NetView platform.

7.3.4.1 View: A Common Interface for Defining Managed Nodes

The View user interface provides a common interface for displaying resources to be managed, and for accessing functions related to those resources. Additionally, related systems management applications are launched from this interface. LAN NetView Monitor 1.0 is one such application.

Under View, systems are grouped together into "management collections." Managed nodes are initially discovered by the topology and discovery services of the LAN NetView Manage product, and put into a single management collection called "All Systems." The user can then subset this "All Systems" collection into smaller management collections through the View interface. Note that the same method of grouping managed nodes is used independent of which management application is being used. The Monitor application is then launched from a management collection. Thus the user of Monitor doesn't have to learn a new interface for defining managed nodes. Nor does the Monitor application need to include its own "node definition and grouping" capability.

7.3.4.2 Synergy with Other Management Applications

Although LAN NetView Monitor 1.0 has much function in and of itself, it gains more function by working in conjunction with the other applications currently offered in the LAN NetView family. As more applications are introduced by both IBM and vendors, there may be even more function that Monitor is able to take advantage of. Current functionality is described below.

- When a threshold is exceeded, the managing system is notified through generation of an alarm. However, with only Monitor running on the managing system, the alarm stops there. If the LAN NetView Tie product is also installed and configured on a managing system, Monitor threshold alarms that the user defines as "critical" can be received by LAN NetView Tie and routed to NetView on the host. Note that LAN NetView Tie does not have to be on the same managing system as Monitor.

- All levels of threshold alarms (not just critical ones) can be received and processed by LAN NetView Fix, as well as by Monitor. Additionally, "quality of service" alarms (such as "disk full" and "log full") can also be handled by LAN NetView Fix.

- Monitor provides a command line interface enabling remote unattended operation for the functions of policy management (creation, starting and stopping), and log file transfer. However, in order to execute these commands on a remote system, a remote command line capability is needed. The LAN NetView Manage product provides such a capability.

  This opens the door for support of a tiered "Manager of Managers" (MOM) solution. In order to support a large number of nodes, it may be best to distribute performance management of these nodes across several managing systems. (This is due to the possible database requirement of collecting data from a large number of systems. The amount of database disk space required will depend on the number of resources being monitored and the frequency of data collection.) However, to minimize administrator resources, it is likely that "management of the managers" will be done from a single, central point. This can be done in the following way:

- Systems to be managed would be divided up among the "first tier" of managing systems. Each managing system would have its own local database to handle its set of managed systems. Policies would be created on each "first tier" managing system and could be created from the "MOM" system through the use of Monitor commands and the Remote Command Line facility of the LAN NetView Manage product. The policies would be set up for automatic log file transfer back to each managing system's database. Policies would then be started and stopped via the command line from the "MOM" system.
- Graphing and reporting could be accomplished from the "MOM" system through remote access to each "first tier" database, as long as the MOM system didn't try to write to these databases (write-access to these databases could be prevented through setup at the MOM system). Since the graphing and reporting facilities only need read access to the database to accomplish their tasks, the central administrator would have access to the data both graphically (realtime and from the database) and through reports.

7.3.4.3 Standardized Interfaces Facilitate Extension to Other Systems

LAN NetView Monitor Version 1.0 is based on the Open System Interconnection (OSI) managing and managed system model, and is written to the X/Open Management Protocol (XMP) interface of the LAN NetView Manage product. The monitored performance resources are modeled into GDMO objects (Guidelines for Definition of Managed Objects), and the performance information for these resources is accessed via the object interface of XMP. As a result, system dependent interfaces to performance data are shielded from the object layer, and hence from the Monitor application itself.

The result is that LAN NetView Monitor 1.0 is easily extendible to the support of other platforms, such as the Novell or AIX platforms. Once agents that conform to the XMP and GDMO object interfaces are written for these systems, Monitor should be able to handle performance information from these systems.

7.3.4.4 Communications Transports Provided by the LAN NetView Platform

As an application written to the XMP interface of Manage, Monitor does not have to be concerned with the underlying management and communications protocols — this is all handled by XMP. Hence, as XMP is extended to support more protocols, Monitor can take advantage of this support. The protocol used by the performance agent is CMIP. CMIP is currently supported over TCP/IP and Logical Link Control (LLC) communications protocols.

This is a difference over the remote support provided with SPM/2 2.0. SPM/2 uses the remote named pipe support of the IBM LAN Server and Requester products, and thus requires that these products be installed. Additionally, due to the security checking required by LAN Server when creating remote named pipes, SPM/2 2.0 requires that both the managing and managed systems be logged onto the domain. LAN NetView Monitor does not require any such logons.

7.3.5 Other Enhancements over SPM/2 2.0

Other enhancements over SPM/2 2.0 that are provided with LAN NetView Monitor are described below. Many of these functions also lend themselves to an administrator environment.

7.3.6 Multiple Monitoring Sessions

Unlike SPM/2 2.0, more than one monitoring session can be active at a single managed system. In "LAN NetView Monitor terms" this means that multiple policies can be active on the same managed system. Each active policy can be initiated from a single managing system, or different managing systems. Each policy at the managed system is totally independent — each can be collecting the same or different data, and any set of thresholds can be active.

7.3.6.1 Additional Data Formatted into Reports

In SPM/2 2.0, although LAN Server and Requester data can be *collected*, this data is not formatted into summarized reports — rather, it is only available through the "dump" format. In LAN NetView Monitor 1.0, there are specific tables set aside in the database for LAN Server and Requester performance data, and this information is included in the standard set of Monitor reports.

Additionally, since data is stored in a database, the user has the capability to provide summarized reports on user-metrics more easily than is possible with SPM/2 2.0. With SPM/2 2.0, dump reports would have to be formatted by writing a program that parses the ASCII file. With Monitor, this data can be summarized through the SQL interface.

7.3.6.2 Graphing Enhancements

The graphing facility of the LAN NetView Monitor product provides many enhancements over the SPM/2 2.0 graphing facility, as described below:

- Metrics from multiple workstations can be displayed in a single instance of the graph.
- Multiple graphs can be active simultaneously, independent of the number of systems being monitored or the number of policies active. The user can define which set of metrics they want to display in each graph instance — any combination of metrics from active policies is possible.
- Any metric can be graphed at the workstation level. SPM/2 2.0 only graphs CPU, Physical Disk, RAM and Page In/Out activity.
- Historical graphing is from the database, rather than from a log file, as with SPM/2 2.0. Given the remote access capability of the OS/2 Database Manager product, graphing from database can be performed from any database client.
- The user can specify the line color, width, and shading for each metric displayed.
- While the graph is active, the user can temporarily disable viewing of various portions and lines of the graph. For example, if a large number of metrics are being displayed, and the user wishes to remove some clutter from the graph in order to get a better look at a couple of metrics of interest, viewing of other metric lines can be temporarily "turned off".
- The graph legend has been enhanced to show more detail about each metric line, including the policy and node the line is from, and line's minimum, maximum, and average values since the graphing session started.

7.4 Summary

SPM/2 2.0 and LAN NetView Monitor 1.0 both handle collection of performance data from systems using the OS/2 operating system. Although there is some overlap in capability between the two products, each offers unique function that makes is a viable product for its own target audience:

- SPM/2 2.0 aids in the developer/analyst and small LAN environment by providing the data and reporting needed for problem identification and resolution. These functions help the developer or analyst delve more deeply into the system for the purpose of problem determination.
- LAN NetView Monitor 1.0 aids the system administrator by not merely collecting data, but by automating data collection, alerting the administrator when thresholds are exceeded, and centralizing collected data into a common database for trend analysis.

Chapter 8. Using IBM LAN NetView Manage Version 1.0 and IBM LAN Network Manager Version 1.1 Together

Managing both the systems and the network aspects of your distributed LAN system requires an understanding of the communications facilities, the operating systems, and the subsystems involved. The IBM LAN NetView family of products focuses on the operating systems and subsystems of the workstations in the distributed LAN system, while IBM LAN Network Manager Version 1.1 focuses on the token-ring communications facilities. This paper provides detailed technical advice on how to use these products together to manage your whole distributed LAN system.

8.1 Introduction

The distributed LAN system is a complicated system. It is made up of many components, including software, processors, and communications facilities. Problems can crop up in any of these components that affect the entire system. To deal with these problems, you need to be able to manage all of the components.

The IBM LAN NetView family of products focuses on the software and the processors in the distributed LAN system, while the IBM LAN Network Manager Version 1.1 product focuses on the token-ring communications facilities of your installation. Used in concert, these products help you to manage both the systems and the network aspects of your distributed LAN system.

8.2 Deciding What's Needed

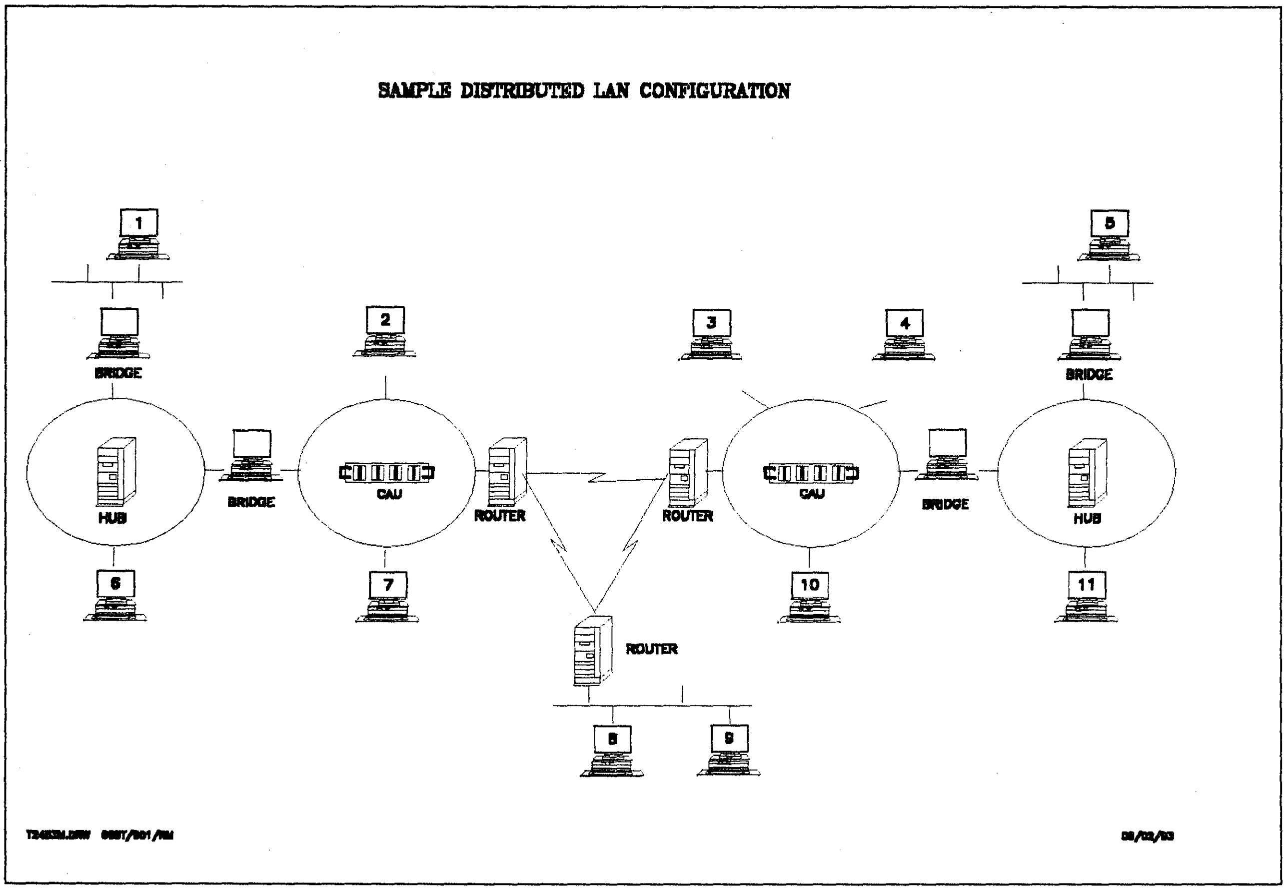


Figure 3. A Sample Distributed LAN System

Suppose your distributed LAN system looks something like the sample shown in Figure 3. There are two bridged token-ring networks and a Ethernet network. These networks are joined by a router network. Hubs and controlled access units (CAUs) support the LAN communications. You have workstations significant to your distributed LAN system on every segment of the token-ring and Ethernet networks. These workstations use OS/2 or DOS (with or without Microsoft Windows). Some may have IBM LAN Server, IBM Database Manager/2, or IBM Communications Manager/2. Others may be Novell Netware servers. Most are clients of some form or the other.

The IBM LAN NetView family of products and the IBM LAN Network Manager product can help you manage such a distributed LAN system. These products must be used in combination, though, to get the complete coverage that you need.

8.3 What the IBM LAN NetView Family of Products Manages

The IBM LAN NetView Manage Version 1.0 product provides the basis of the managing workstation built around the IBM LAN NetView family of products. The IBM LAN NetView Manage product provides access to any CMIP (Common Management Information) or SNMP (Simple Network Management Protocol) object, if the object is accessible across a bridged LAN or via TCP/IP communications. The IBM LAN NetView family of products provides CMIP representations of the workstation hardware (processor, adapters, display, keyboard, etc.), the OS/2, DOS, and DOS with Windows operating systems, and the OS/2 subsystems (IBM LAN Requester for OS/2, IBM LAN Server, IBM Database Manager/2, and IBM Communications Manager/2). Various vendors (including IBM) provide SNMP representations of their hubs, routers, and other communications equipment.

Within a bridged LAN, the IBM LAN NetView Manage product may use the CMIP over Logical Link Control (CMOL) management protocol, the CMIP over TCP/IP (CMOT) management protocol, or the SNMP management protocol. Across a router, the CMOT or SNMP management protocols may be used. Also, through a connection to Novell's management products, objects accessible via IPX may be accessed.

Up to 1024 simultaneous communications sessions may be supported, although the limit will be lower based on the constraints of the processor running IBM LAN NetView Manage. Typically, 40-250 workstations are managed from a workstation installed with the IBM LAN NetView Manage product. The applications that run atop of IBM LAN NetView Manage may be more selective in which objects they support. For example, the IBM LAN NetView Monitor Version 1.0 product provides performance monitoring functions for OS/2 operating systems, IBM LAN Requester for OS/2s, and IBM LAN Servers.

In the sample distributed LAN system shown in Figure 3 on page 110, any of the numbered workstations can act as a managing workstation with the IBM LAN NetView Manage product installed. Using the CMOT management protocol, the managing workstation would be able to manage all of the CMIP objects in any of the workstations in the distributed LAN system. For those CMIP objects on the same bridged LAN as the managing workstation, the CMOL management protocol could be used instead. This presumes that either IBM LAN NetView Manage, IBM LAN NetView Enabler Version 1.0, or IBM LAN NetView Agents for DOS Version 1.0 are installed on the numbered workstation, with IBM LAN NetView Agents Extended installed where appropriate. Using SNMP, any of the SNMP objects throughout the distributed LAN system could be managed.

Depending on your needs, more than one managing workstation could be defined. You can choose whether to subset what each managing workstation is managing to avoid overlap, or you can have each managing workstation manage all of the same objects. In this second case, you will need to establish procedures to recognize that more than one managing workstation may be acting upon a particular object at the same time.

8.4 What the IBM LAN Network Manager Product Manages

The IBM LAN Network Manager product manages token-ring segments, broadband PC Network segments, baseband PC Network segments, and adapter cards attached to the token-ring segments. These objects are accessed via the IEEE standard 802.1 and 802.5 protocols. The IBM LAN Network Manager product also manages LAN bridges, using a private protocol. Controlled access units (CAUs) are managed using an early version of the CMOL management protocol. Likewise, the IBM LAN Network Manager product uses CMOL to manage objects represented by the LSM Version 1.0 (LSM) product, which provides access to the "upper" layers of the adapter card and to some of the asset information about the workstation on which it is running.

An IBM LAN Network Manager can manage up to 256 bridged segments (255 bridges). Up to 1000 objects within these segments can be classified as "critical resources" that are monitored very closely. IBM LAN Network Manager must be attached to one of the bridged segments, since it cannot access objects that are not within the bridged LAN. At a time, one IBM LAN Network Manager workstation may manage a given object, while up to three other IBM LAN Network Manager-installed workstations may be monitoring the given object. These observers cannot request actions on the object. Configuration options for IBM LAN Network Manager define which objects are managed as critical resources, which objects are managed as normal resources, and which objects are observed.

In the sample distributed LAN system in Figure 3 on page 110, workstations 2, 3, 4, 6, 7, 10, or 11 may have the IBM LAN Network Manager product installed, since these are the workstations connected to token-ring segments of bridged LANs. Installing the IBM LAN Network Manager product on workstation 4, for example, provides coverage for the two token-ring segments and the Ethernet segment of the bridged LAN, the two bridges connecting these segments, the CAU supporting one of the token-ring segments, and the token-ring adapter cards throughout the bridged LAN. If workstation 3, 5, 10, or 11 have the LSM product installed, the managing workstation would be able to manage the upper layers of those workstations' adapters and the processors of those workstations.

Workstations 2, 6, or 7 could be chosen for a second managing workstation with the IBM LAN Network Manager product installed. This workstation would cover bridges, CAUs, and segments to the left of the router network. Also, it would handle any workstations with the LSM product installed in the left bridged LAN.

8.5 Combining Coverage

As you can see, neither the IBM LAN NetView Manage product nor the IBM LAN Network Manager product provide access to all of the objects within your distributed LAN system by itself. Together, you can achieve far more complete coverage.

When you put both the IBM LAN NetView Manage product and the IBM LAN Network Manager product into the same bridged LAN, either product can access objects visible via the LSM product. Since the IBM LAN NetView Manage and IBM LAN NetView Enabler products may be configured to use some of the same code as used by the LSM product, processor information regarding workstations with either of these products installed will be visible to the IBM LAN Network Manager product as well. Likewise, processor information about workstations with the IBM LAN NetView Agents for DOS product installed will be available to the IBM LAN Network Manager product.

You have two options for combining the coverages provided by the IBM LAN NetView Manage and the IBM LAN Network Manager products. You may choose to have both products coresident in a single managing workstation or you might distribute the products across two managing workstations. For either approach, you will want to access information from either product easily.

In the sample distributed LAN system, you might choose to install the IBM LAN NetView Manage product in workstations 2 and 3. You might put the IBM LAN Network Manager product in workstation 2 (making it a coresident managing workstation) and in workstation 4 (a distributed managing workstation). Suppose workstation 2 is responsible for the left bridged LAN. Workstations 3 and 4 might be responsible for the router network, the Ethernet network, and the right bridged LAN. Workstation 7 might be a critical server that needs to be managed by both workstation 2 and the workstations 3 and 4. With this division of responsibilities in mind, we can look at the considerations for using a coresident approach versus a distributed approach.

8.6 Considerations

Based on the approach that you choose, there are some items to consider while installing and operating the IBM LAN NetView Manage product and the IBM LAN Network Manager product together. In addition, there is an item to consider whenever you use both of these products in a distributed LAN system, independent of the approach you choose.

8.6.1 Coresident Managing Workstations

The CMOL management protocol is defined to allow a single Service Access Point (SAP) in the Logical Link Control protocol layer. This means that only one managing program may attach to CMOL in a single workstation. Since the IBM LAN NetView Manage product and the IBM LAN Network Manager product can be configured to use the CMOL management protocol, using these products in the same workstation has some important considerations.

At installation time, the IBM LAN NetView Manage product requires some configuration of the Heterogeneous LAN Management (HLM) device drivers that implement the CMOL management protocol. The IBM LAN Network Manager product uses the default configuration. Installing the IBM LAN Network Manager product after the IBM LAN NetView Manage product is installed would cause the default configuration to be restored. This would hamper the IBM LAN NetView Manage product's operation. Installing the IBM LAN Network Manager product first, then the IBM LAN NetView Manage product with the CMOL management protocol stack selected, avoids this problem.

When starting these products together, the order in which you start will affect what each product can manage. If the IBM LAN Network Manager product is started first, it will attach to the CMOL management protocol. When the IBM LAN NetView Manage product starts, any objects which it needs to access via CMOL will be unavailable since IBM LAN Network Manager attached first.

If the IBM LAN NetView Manage product is started first, the IBM LAN Network Manager product would not be able to access the CAUs or the LSM-supported objects. Fortunately, the LSM-supported objects are CMIP objects and may be managed from the IBM LAN NetView Manage product like all other CMIP objects. However, the object-specific display functions provided by the IBM LAN Network Manager product for these LSM-supported objects are not present in the IBM LAN NetView family of products. Unfortunately, the pre-standard version of CMOL used to manage the CAUs is incompatible with standard CMOL. Consequently, the CAUs cannot be managed from the IBM LAN NetView Manage product.

If you want to use the CMOL management protocol for some or all of the objects managed from the IBM LAN NetView Manage product, you will need to start the IBM LAN NetView Manage product first and take advantage of the general facilities in the IBM LAN NetView Manage product to handle the LSM-supported objects. Alternately, you can configure the IBM LAN NetView Manage product to not use CMOL and start both products in either order, using the full functions of both products. Finally, if you want to use CMOL from the IBM LAN NetView Manage product and to manage the CAUs and LSM-supported objects from the IBM LAN Network Manager product, you will need to switch to the distributed approach (see 8.7, "Considerations for Distributed Managing Workstations").

Returning to our sample distributed LAN system, workstation 2 has both the IBM LAN NetView Manage product and the IBM LAN Network Manager product installed. If workstations 1 and 6 are DOS workstations with the IBM LAN NetView Agents for DOS product installed, the only way to manage those workstation is via CMOL from the IBM LAN NetView Manage product. This requires that the IBM LAN NetView Manage product be started first. The contents of workstations 1 and 6 would be managed by the IBM LAN NetView Manage product, along with the hub (using the SNMP protocol). The IBM LAN Network Manager product would manage the two token-ring segments, the Ethernet segment, and the two bridges in the left bridged LAN. Since the IBM LAN NetView Manage product is started first, neither it nor the IBM LAN Network Manager product would have access to the CAU.

8.7 Considerations for Distributed Managing Workstations

Using the distributed approach avoids the problem of which program attaches to the CMOL management protocol first. the information known by one product while using the other product. There are two ways to resolve this.

First, you could resort to placing the consoles from the different workstations side-by-side. This works in situations where the geography of your distributed LAN system allows you place the consoles this way. In our sample distributed LAN system, you could use this approach to combine the information from workstation 3 (with the IBM LAN NetView Manage product installed) with information from workstation 4 (with the IBM LAN Network Manager product installed). However, since workstation 7, which is attached to the left bridged LAN, is also within the responsibility of the IBM LAN NetView Manage installation on workstation 3, you will need to access the IBM LAN Network Manager product installed on workstation 2 in order to understand the health of the token-ring segment to which workstation 7 is attached. Since workstation 2 must be attached to the left bridged LAN, which is located at some distance from workstation 3, this option does not work.

The second approach is to use a remote console product to access one workstation from another. IBM offers the IBM Distributed Console Access Facility Version 1.1 (DCAF) product, which allows an operator to take over the display and keyboard of a remote workstation. This version of DCAF supports Presentation Manager screens and mouse interactions. Workstation 2 could be accessed via DCAF from workstation 3 when access to information about the health of the token-ring segment is needed. In this way, any managing workstation configured for use with DCAF could be accessed by any other managing workstation as you need to gather information about your distributed LAN system.

When using the IBM LAN Network Manager product's graphical display, any workstation with either the IBM LAN NetView Manage or IBM LAN NetView Enabler products installed will be flagged as a managing workstation. This is due to the way in which the IBM LAN NetView Manage and IBM LAN NetView Enabler products announce themselves to the CMOL management protocol.

8.8 General

Regardless of the approach you select, there is one other consideration. Any workstation that uses CMOL, whether managing or managed, needs to be using a level of the HLM device drivers that is equal to or later than the level shipped with the IBM LAN NetView family of products. Thus, all workstations where the IBM LAN Network Manager product or the LSM product are installed need to be checked. Since the proper level of HLM device drivers is installed with the IBM LAN NetView Manage, IBM LAN NetView Enabler, or IBM LAN NetView Agents for DOS products, any workstation where these products are installed will not have to be checked.

8.9 Summary

Managing all aspects of a complex distributed LAN system like the one shown in Figure 3 on page 110 requires the use of at least two workstations with the IBM LAN Network Manager product installed and one or more workstations with the IBM LAN NetView Manage product and associated applications installed. Although there are some considerations when installing these products together, a little planning is enough to ensure complete management coverage. Using a distributed management workstation approach, the full functions of the IBM LAN Network Manager product and the IBM LAN NetView family of products (tied together through the use of the IBM Distributed Console Access Facility Version 1.1 product) may be applied to the management of your whole distributed LAN system.

Chapter 9. Remote LAN Management Using IBM LAN NetView

The movement toward distributed processing using personal workstations connected via local and wide area networks (LANs/WANs) is bringing more focus to the area of systems and network management. In many enterprises, where once an IS (Information System) organization provided an end-to-end solution for management of the centralized mainframe and the subnets connected to it, much of the systems management, in terms of backup/restore, software distribution, inventory of assets, and problem management as some examples, must be done remotely. The IBM LAN NetView family of products provides a low-cost solution for the management of personal workstations and network components. In particular, it provides features which give LAN administrators the ability to remotely manage their LAN resources. This paper discusses some of the problems enterprises are facing in the area of network management, and describes the IBM LAN NetView family of products. The use of related IBM products, as a means to provide remote LAN management solutions is also described. It is aimed at service providers and LAN administrators, both of whom can implement remote LAN management solutions using these products.

9.1 Introduction

The costs associated with local area network management--keeping LANs operational and performing administrative functions, are getting a lot more attention these days. Companies are focusing more than ever on their Information Systems expenditures, and the expected savings from downsizing to LANs appears to be mitigated because of these costs.

Why is the cost higher than expected? What can be done to control the escalating cost of administration, help desk, and software distribution functions? Part of the answer lies in providing effective tools for remote management of the LAN resources. IBM LAN NetView provides excellent capabilities in the area of remote systems management. In describing this support, we will look at the LAN NetView family of products, including the LAN NetView Monitor and LAN NetView Fix applications, and the LAN Distance* and DCAF (Distributed Console Access Facility) products, which add significant remote management capabilities to an overall remote management solution.

9.2 The Cost of LAN Management

What does it really cost to run corporate local area networks? There's no easy answer. However, LAN users are becoming more and more aware that a significant portion of the cost of running LANs is hidden. Unlike the obvious costs associated with buying and installing the cabling, hardware, and software, these hidden costs have been difficult to quantify. The basic premise upon which the decision to migrate from a mainframe environment to a distributed LAN environment is that the hardware is much less expensive. What's often overlooked is the fact that the hardware and software requires extensive technical support and administration. People costs are high. The cost of LAN administration is constantly going up. Not only in terms of LAN administrators' remuneration for keeping the system operational and performing functions like adding users, distributing software, and controlling access to LAN resources, but also in the time spent by end users themselves; performing functions like backing up/restoring their systems, maintaining private libraries, etc. Thus the cost, like the systems and network management itself, is distributed throughout the enterprise, and cannot easily be accounted for.

From a networking standpoint, a SNA network generally costs less to support because of the smaller staff needed to centrally manage the nodes. By comparison, LANs require more administrative and technical staff to provide the same level of support. The reason for this is the distributed nature of the workgroup LANs. Though some network management tools allow monitoring of the LANs, often changes must still be made locally.

9.3 Controlling the Cost

The keys to controlling LAN costs are: controlling the ratio of support people to users, automating LAN management, using standard hardware and software, and in some cases--selective outsourcing. Using tools like smart hubs, protocol analyzers, and network management systems can yield significant savings through automation of LAN management.

9.3.1 The LAN Support Staff

With LAN users and applications increasing steadily, the LAN administrator and technical support people often find themselves bogged down with their daily tasks of adding and changing userids, profiles and applications, installing equipment, and inevitably responding to crisis situations.

As more and more applications are integrated at the LAN level, often a Wide Area Network configuration results, increasing the complexity of the network, and burdening the technical support staff with more complex problems. Combine this with the budgetary constraints most companies are enduring these days in terms of limited hiring, and you have the potential for a degradation in network service and performance.

One of the keys to controlling the costs mentioned above was automating LAN Management. There are several network management systems on the market today that offer tools that can assist the LAN administrator and support staff by performing automated LAN (and WAN) management. These tools, usually classified as systems and/or network management applications, can be used for:

- Performance management
- Problem management (fault management)
- Asset management
- Configuration management
- Security management
- Software distribution

Certainly all of these functions are not required in all cases, but the point is that a number of tools exist which can provide the support staff with the means to respond to the demands of their LAN users.

9.3.2 Outsourcing

More companies these days are taking a strong look at using external LAN service and support vendors as a viable solution for their network administration. The primary reason for this has to do with with gaining experienced network support in emergency situations for a reasonable cost, with no additional staffing. This means no outlays for benefits such as health insurance and vacation time.

When you're thinking about outsourcing your LAN management services you would normally weigh the cost of in-house management vs. what the service provider is going to charge you. However, one also needs to consider the value of the network administrator's time to perform tasks other than the maintenance functions mentioned previously, and the increased value of the network to end users if downtime is reduced. Downtime often translates into a major expense in lost productivity.

In addition to administrative and technical support functions, external LAN service providers can help in planning network changes and consulting on network design.

9.4 Remote LAN Management

Where LANs are geographically distributed within an enterprise, remote network management tools are vital to keeping LAN costs down. Whether it's a service provider or an in-house support staff maintaining the LAN, costs incurred with travelling to remote locations for problem determination, software installation, or to simply apply a program fix, can be significant. This expense can be lessened, and even avoided in many cases by using good remote LAN management tools.

These tools are generally applications or systems/network management platform services which enable an administrator or technical support person to remotely access a *managing system*, which in turn can be used to monitor and control the network resources. In particular, this paper will describe the IBM LAN NetView family of products and other IBM applications which can be used in conjunction with the LAN NetView products to provide the means for remote LAN management, either through a remote dial-up capability, remote console access, or programmatically.

9.4.1 LAN NetView

The IBM LAN NetView family of products provides a standards-based platform for the creation and implementation of systems management applications for the LAN workgroup environment. The management framework for the products is an OS/2 implementation of technology selected by the Open Software Foundation (OSF) for the Distributed Management Environment (DME). The platform services and applications allow the management of LAN-attached clients and servers running the OS/2 2.x, IBM DOS 5.0 and 6.1, Microsoft DOS 5.0 and 6.0, and DOS with Microsoft Windows 3.0 or 3.1 operating systems.

Within the IBM NetView family of products (NetView, NetView/6000, and LAN NetView), LAN NetView is positioned as the systems management platform of choice for the small to medium size enterprises where the customer's installed base is primarily personal workstations (for example PS/2* and compatibles), and the number of network nodes to be managed is less than 500, per managing system.

The LAN NetView platform architecture is based on the OSI (Open Systems Interconnection) Systems Management Model, which defines:

1. A managing system with the following attributes:
   - Has a user interface.
   - Executes the management applications.
   - Performs a standard set of actions on managed objects.
2. A managed system on which an agent or agents monitor and control managed resources and provide notifications of events to the managing system.

The managing system in a LAN NetView configuration would have the LAN NetView Manage version 1.0 product and agents installed on it. Any managed OS/2 systems would have LAN NetView Enabler version 1.0 and agents installed, and DOS systems (as well as DOS with Microsoft Windows 3.0 or 3.1) would have LAN NetView Agents for DOS installed. Remote management of the LAN resources is enabled by LAN NetView's ability to accept commands from a command line or application program running on the Managing system, from NetView via SNA connection, from another OS/2 system with the Distributed Console Access Facility (DCAF) installed, and/or from a remote system asynchronously connected to the managing system using the IBM LAN Distance product. These remote management facilities are described in succeeding subsections.

There's a substantial value-add here that the LAN NetView product brings to LAN management; it's a platform for systems management AND network management. You'll see some of the systems management capabilities in the example scenarios later on in the paper, such as recognizing the "Disk nearing capacity" condition on the LAN Server, or the use of LAN NetView Monitor to gather performance data. Network management is generally thought to refer to the management of SNMP devices, such as bridges, routers, hubs, etc. The LAN NetView platform can be used for managing these devices as well, since the platform supports SNMP as well as CMIP. The Request Manager applet (mini-application), which is included in the LAN NetView Manage 1.0 product, can be used for this purpose. This will support querying and controlling SNMP devices at the MIB II level. Since objects are maintained in the LAN NetView metadata database in GDMO (ISO Guidelines for the Definition of Managed Objects) format, SNMP SMI-formatted MIB extensions cannot be easily compiled into the metadata database for use by management applications. (It can be accomplished by converting the SNMP MIB into GDMO format and then compiling the output into the metadata database using the LAN NetView metadata compiler.) However, the IBM 6611 multiprotocol router MIB and the RMON (Remote MONitoring) MIBs (for Token-Ring and Ethernet) are included in the LAN NetView Manage 1.0 product. In addition, IBM is pursuing a post-release electronic distribution of the more popular vendor MIBs. More comprehensive support for SNMP device management is anticipated in a subsequent release of LAN NetView products.

Note: For a complete description of the LAN NetView platform and the LAN NetView family of products, refer to the paper entitled "OS/2 Distributed Systems Management LAN NetView Family of Products".

9.4.1.1 Remote Command Line Interface

One of the mini-applications included with the IBM LAN NetView Manage version 1.0 product is the Remote Command Line Interface (RCLI). It provides the ability to invoke commands at managed OS/2 workstations (with LAN NetView Enabler version 1.0 running), from a managing workstation where LAN NetView Manage version 1.0 is running. A remote command can be issued from either an OS/2 command line, or via the LAN NetView Manage graphical user interface called View. Regardless of the method of invocation, commands must be non-interactive-- no user input is required once the resulting program begins execution. This remote command line support can be used by the LAN administrator to administer personal workstations (where LAN NetView Enabler version 1.0 is running) from:

- An OS/2 prompt on a LAN NetView managing system.
- A host NetView console, through a LAN NetView managing system, using the OS/2 Communications Manager Remote Operations (ROPS) feature.
- A program running on a LAN NetView managing system. An example of this would be a fault management application responding to an alarm notification by issuing a remote command to the managed system for corrective action.

This feature of LAN NetView Manage 1.0 could be used by service providers to remotely monitor systems managed by LAN NetView Manage 1.0 (including LAN Servers), and to control their resources from a central host with NetView running, or from a personal workstation where LAN NetView Manage 1.0 is used.

9.4.2 DCAF

The IBM Distributed Console Access Facility (DCAF) allows a user at an OS/2 workstation to access and control the keyboard and display of another OS/2, or DOS workstation. This enables remote network management, administration, and application support of remote systems. The DCAF product supports OS/2 applications as well as most DOS applications. The following connectivity is supported:

- SNA using LU6.2 over SDLC, X.25, or Token-Ring transports
- NetBIOS over Token-Ring
- Switched Asynchronous connection

The DCAF product has two main operating states: monitoring and active. When the DCAF session is in the monitoring state, the controlling workstation user sees a screen image of the target workstation's display. The target workstation user has complete control of the target workstation operations.

When the DCAF session is in the active state, the controlling workstation user operates and controls the target workstation. The keystrokes typed by the controlling workstation user are relayed to the target workstation and acted upon as if they were typed by the target workstation user. The keystroke input and the resulting screen image of the target workstation's display are seen on both the controlling and target workstations. During the active state, keystroke input from the target workstation user is not accepted. However, there is a *hotkey* facility to allow the user at the target workstation to regain control.

Multiple sessions can be established, using OS/2 sessions for each target workstation to be monitored/controlled.

Since DCAF is an OS/2-based application, it can coexist with LAN NetView Manage Version 1.0 on the managing system for use in accessing the managed systems. This gives the LAN support staff the ability to respond to users' needs for technical assistance without having to travel to the remote location. This is extremely helpful for tasks such as remote problem determination and education. The DCAF program can also be initiated from a remote system connected to a LAN NetView managing system, from which the user could perform LAN management functions. In this type of configuration an administrator could provide centralized monitoring and control of the network from a remote site.

DCAF can provide valuable assistance in managing the network through:

- Remote Help Desk assistance for applications, education, and maintenance of applications.
- Remote problem determination for trace and dump analysis, including the transfer of data.
- Remote control of unattended workstations (for example LAN Servers).
- Remote management of Personal System/2* (PS/2) workstations and accessibility to data and programs stored on the workstation.
- Remote access to system consoles when they are implemented on PS/2s.
- Remote monitoring of work in progress on target workstations.

The IBM Distributed Console Access Facility Version 1.1 product is generally available as product number 20G1013.

9.4.3 LAN Distance

When remote LAN management requires a switched connection such as asynchronous, synchronous or ISDN over phone networks, another viable option to consider is the IBM LAN Distance product. The LAN Distance product enables remote users to transparently run their LAN-based applications over these switched connections.

The LAN Distance Connection Server supports up to 32 simultaneous communication ports and provides a full range of configurable security and administrative features. In essence, the LAN Distance product provides the LAN system administrator with effective tools for managing the network.

The LAN Distance product uses the *Remote Node Approach* to provide full addressability to the LAN. The remote node approach entails use of a device driver which enables the LAN-attached communication server to take incoming data off a WAN and put it onto the LAN, and, to take outgoing data off the LAN and put it onto the WAN. In addition to providing transparency and remote LAN management capabilities, this approach allows remote workstations to access distributed LAN-attached servers and peer services. This means that remote workstations can access information and services wherever they reside on the LAN, rather than the LAN having to use a central, dedicated server to accommodate access by the remote workstations.

Remote LAN Access software products generally provide remote machines with the ability to access data on a LAN-attached server using asynchronous connections at rates between 2400 to 14400 bits per second (bps). The LAN Distance product is optimized for higher speed (9600 bps and greater) connections, and includes support for the following LAN and WAN connectivities:

- LAN types
  - Token Ring
  - Ethernet
- WAN types
  - IBM ISDN Basic Rate Adapter
  - Asynchronous Communications Port
  - IBM Dual Asynchronous Adapter
  - IBM RTIC* Portmaster* Adapter (Asynchronous/Synchronous)
  - IBM Wide Area Connector* (Synchronous)

The LAN Distance product supports the following NDIS**-enabled protocols and application programming interfaces:

- NetBIOS
- 802.2
- ODI** requester
- TCP/IP

The LAN Distance product supports any network operating system which resides over an NDIS interface, including IBM LAN Server, Microsoft LAN Manager, and Novell Netware Server. Using the 802.2 protocol called IBM Communications Manager, you can run SNA applications over LAN Distance connections.

Additionally, the LAN Distance product features an extensive set of configurable security options, and provides full administrative support for monitoring connection status as well as logging errors, user data, and audit information.

The LAN Distance product provides an excellent means for a service provider, or a LAN systems administrator to implement remote management of LAN-based resources. There are two approaches to this, similar to the approaches described above in the section on DCAF. First, access to the client managed systems can be gained via dial-up means, and technical assistance can be provided for end users for problem determination and resolution, and education. Secondly, unattended managing systems can be accessed using the LAN Distance product, and management functions for monitoring and controlling the network can be performed remotely. In either case, the LAN NetView family of products can be used effectively in conjunction with the LAN Distance product to allow remote LAN systems management.

9.5 Providing Remote Management Services Using LAN NetView

Often, access to client organizations being managed by service providers is across distances that require a telephone line connection. This emphasizes the need for comprehensive remote management capabilities in the systems and network management system they choose. IBM LAN NetView products help satisfy this need by providing a standards-based platform specifically designed for distributed systems management. When used in conjunction with the DCAF and/or LAN Distance products described above, and with appropriate systems/network management applications from IBM and OEM sources, a service provider or system integrator can produce a viable, low-cost solution for remote LAN management.

Let's look at some examples of how this can be achieved, using some remote management scenarios. For all examples we'll assume that LAN NetView Manage 1.0 is installed on the managing system(s), which also have OS/2 2.x as the operating system, LAN NetView Enabler 1.0 is installed on the OS/2 managed systems, and LAN NetView Agents for DOS is installed on the systems running DOS.

9.5.1 Scenario 1 - Problem Management

LAN NetView Fix 1.0 is a problem management application written to the LAN NetView platform and is designed to receive and process CMIP notifications and SNMP traps in an OS/2 2.x environment.

The configuration for this scenario consists of a central site using a personal workstation with OS/2 2.x, LAN NetView Manage 1.0, and LAN NetView Fix 1.0 running. The workstation is connected to a remote LAN with a configuration of OS/2 systems; a managing system with LAN NetView Manage 1.0 and LAN NetView Fix 1.0 running, a server with OS/2 LAN Server (managed system; remember, LAN NetView Manage 1.0 manages servers and clients), LAN NetView Enabler 1.0, and LAN NetView Agents Extended 1.0 installed, and several clients (all managed systems) with LAN NetView Enabler 1.0 installed. The central site is linked to the remote LANs (assuming there are other LANs being managed) using TCP/IP via telephone line connection. Refer to Figure 4 on page 125, Figure 5 on page 126, and Figure 6 on page 127.

The system administrator at the central managing site wants to be notified of only those alarm conditions which cannot be readily handled by applications on the managing system at the remote site. The LAN NetView Fix application installed at the remote site was setup to automatically process and/or log the less severe alarms. The LAN NetView Fix application installed at the central site is registered to receive higher severity alarms.

As an example of an automated response to an event: One of the printers attached to the remote site's server was found to have changed state to an offline condition. This is detected by the LAN NetView OS/2 agent and the event is routed to the LAN NetView Fix program. Since this area is unattended, the user-defined action that LAN NetView Fix 1.0 takes is to call a pager number and log the event condition, so someone will respond to the alarm, read the indication, and fix the printer.

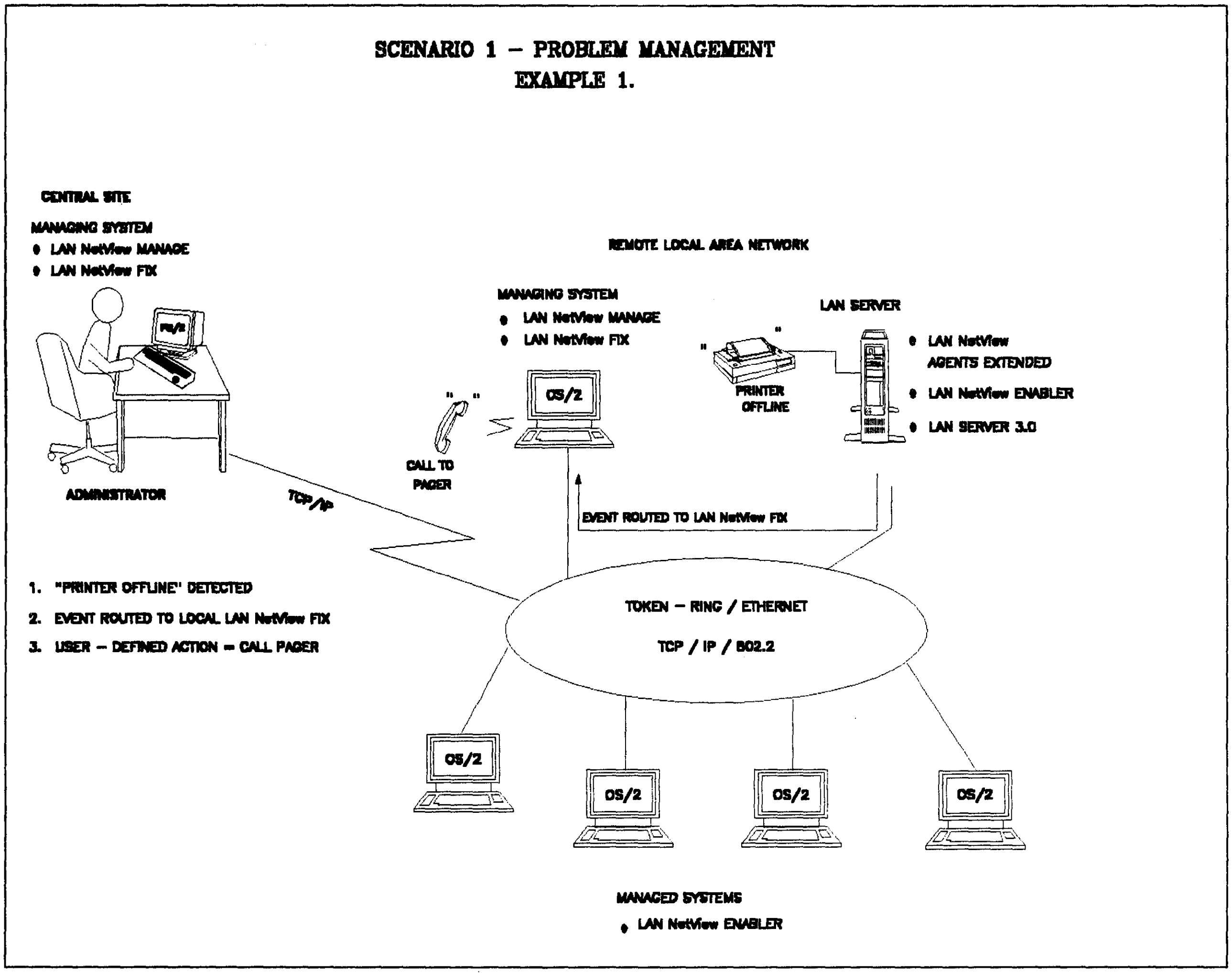


Figure 4. Scenario 1 - Problem Management -- Example 1

Another example would be: a change is detected by the LAN NetView OS/2 agent on one of the client workstations which indicates the system's CONFIG.SYS file was changed. (files such as CONFIG.SYS or .INI files are, in our example, monitored for activity because of their implied system configuration changes). An event notification is sent to the LAN NetView Fix application at the remote site. The program takes action based on the user-defined action; in this case a command is sent back to the client to issue a system shutdown and re-boot, so that the newly-modified CONFIG.SYS can take effect.

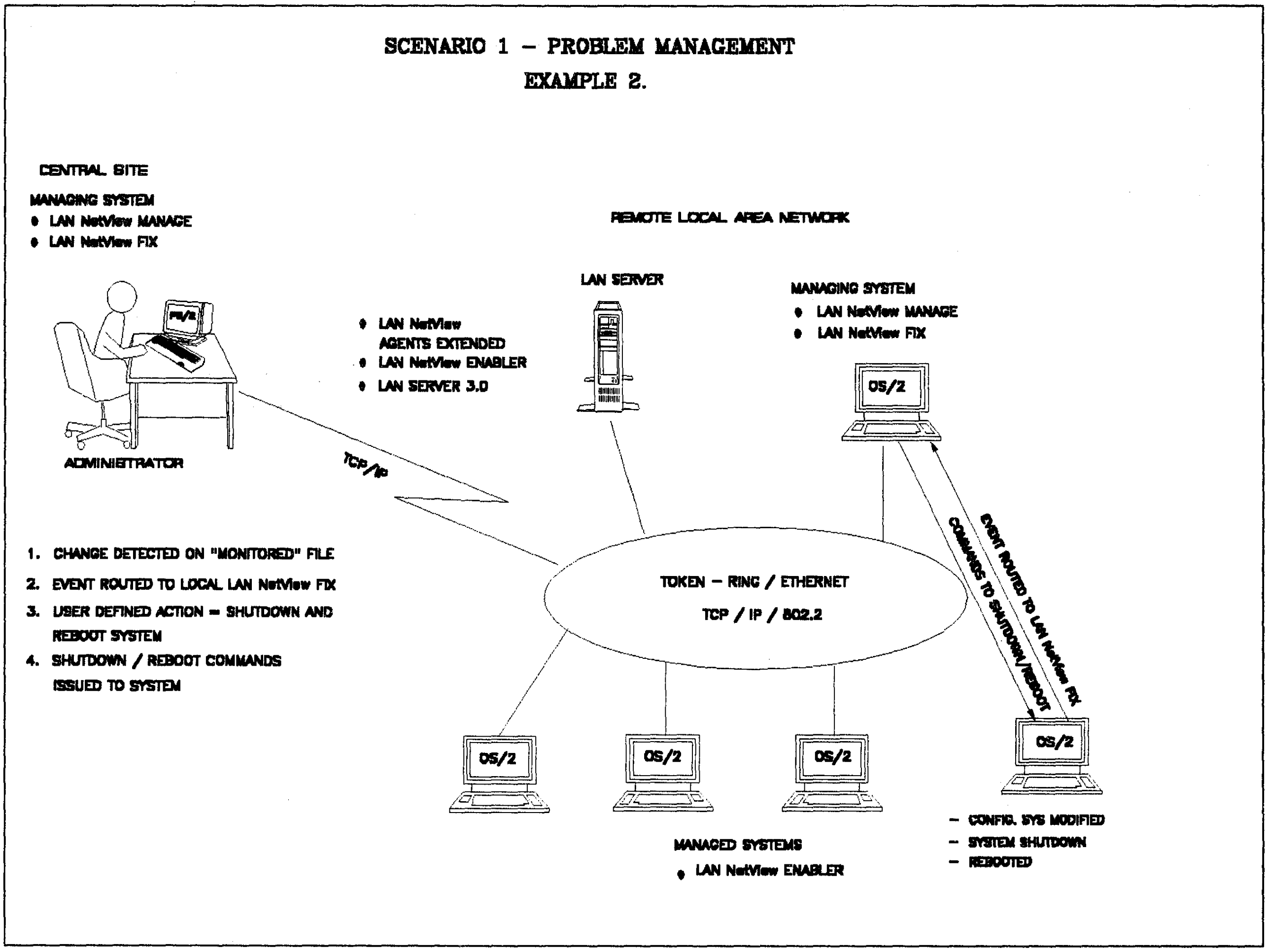


Figure 5. Scenario 1 - Problem Management -- Example 2

As an example of a higher severity alarm: the LAN NetView LAN server agent detects a "disk nearing capacity" condition on the server at the remote site. In this case, the LAN NetView Fix application on the central managing system is registered for this particular alarm and the notification is thus routed to the administrator at the central site. Once the system administrator sees this indication from his/her LAN NetView Fix 1.0 Event Console, he/she can, for example, make use of the Remote Command Line Interface of LAN NetView Manage 1.0 to issue a CHKDSK to the remote server to determine if further action is required, and if so, possibly switch to an alternate server while files from the primary server are archived and disk space re-claimed.

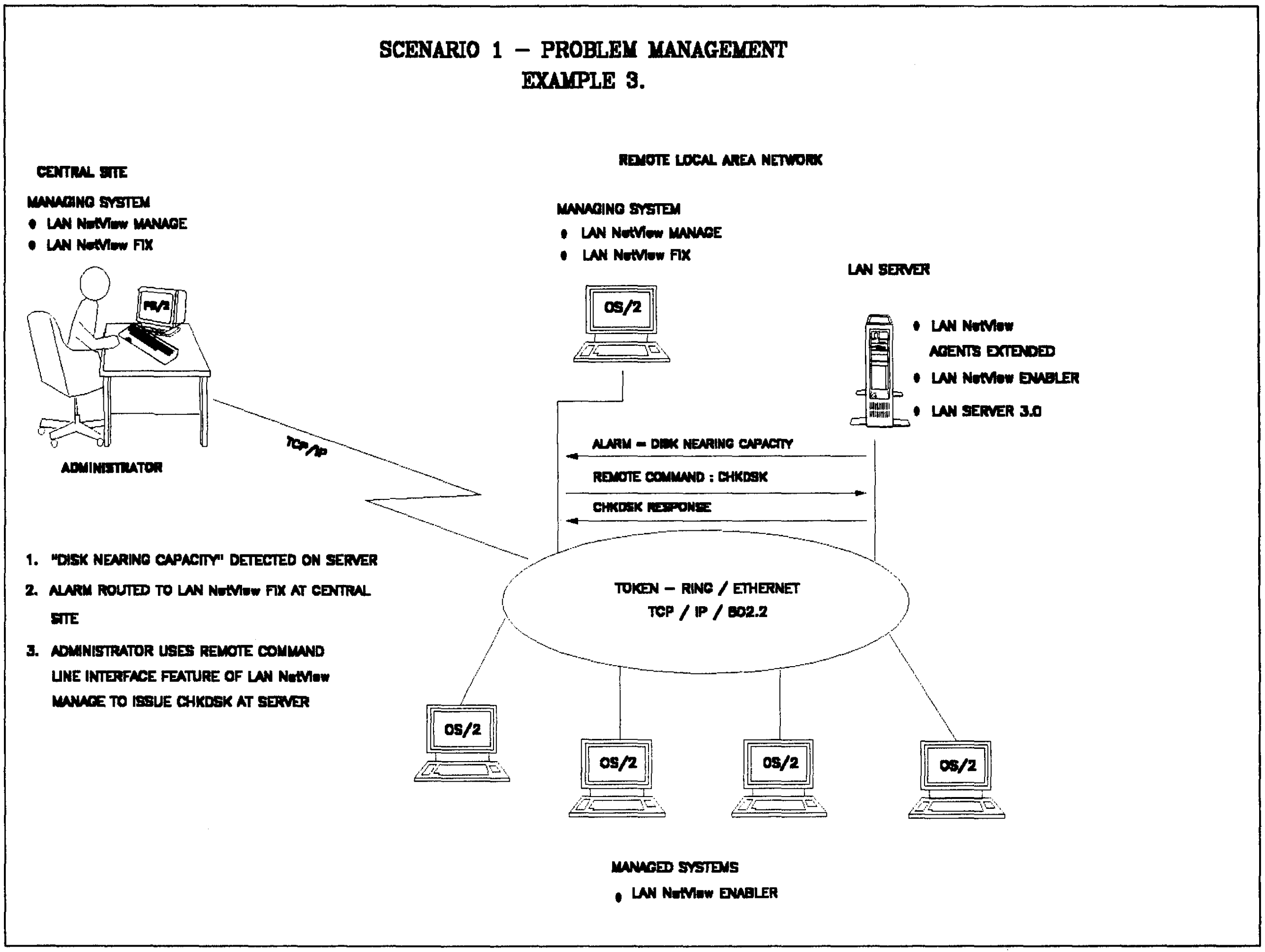


Figure 6. Scenario 1 - Problem Management -- Example 3

9.5.2 Scenario 2 - Monitoring Server Performance

Since the server is vital to LAN processing, it's important to maintain performance, in terms of CPU, memory, and DASD utilization. LAN NetView Monitor 1.0 is an application that provides comprehensive performance management of OS/2 systems through its data collection, graphing, threshold monitoring, and reporting functions.

The configuration for this scenario consists of a central site using a personal workstation with OS/2 2.x, LAN NetView Monitor 1.0, and the LAN Distance product running. The workstation is connected to a remote LAN with a mixed configuration of OS/2 and DOS systems, a LAN Distance Server, a LAN server with OS/2 LAN Server, LAN NetView Enabler 1.0, and LAN NetView Agents Extended 1.0 installed, and several clients (all managed systems) with LAN NetView Enabler 1.0 installed on the OS/2 systems and LAN NetView Agents for DOS on the DOS systems. The central site is linked to the remote LAN Distance server via telephone line connection. Refer to Figure 7 on page 128.

In this case the system administrator at the central site wishes to gather data for capacity planning, while at the same time monitor the DASD usage at the server. He/she uses the LAN Distance program to access the remote LAN and, at the managing system, invokes the LAN NetView Monitor program from the LAN NetView Manage 1.0 graphical user interface called View. The administrator has previously defined a performance *policy* for the server to be monitored. This defines things such as threshold levels, monitoring schedule, data collection interval, etc. Once this policy is started, a performance agent at the server (remember this is a managed system) is started, and performs actions based on the policy settings (for example, collecting and logging data according to schedules/intervals). When the administrator wants to see the current DASD utilization, he/she uses the graphing facilities of LAN NetView Monitor 1.0, to bring up a realtime graph, representing the metrics for this data.

Since all performance data can be stored in either an IBM Extended Services or DB2/2 database, the administrator can generate utilization, trend analysis, and workload reports to assist him/her in capacity planning and performance analysis at the central site.

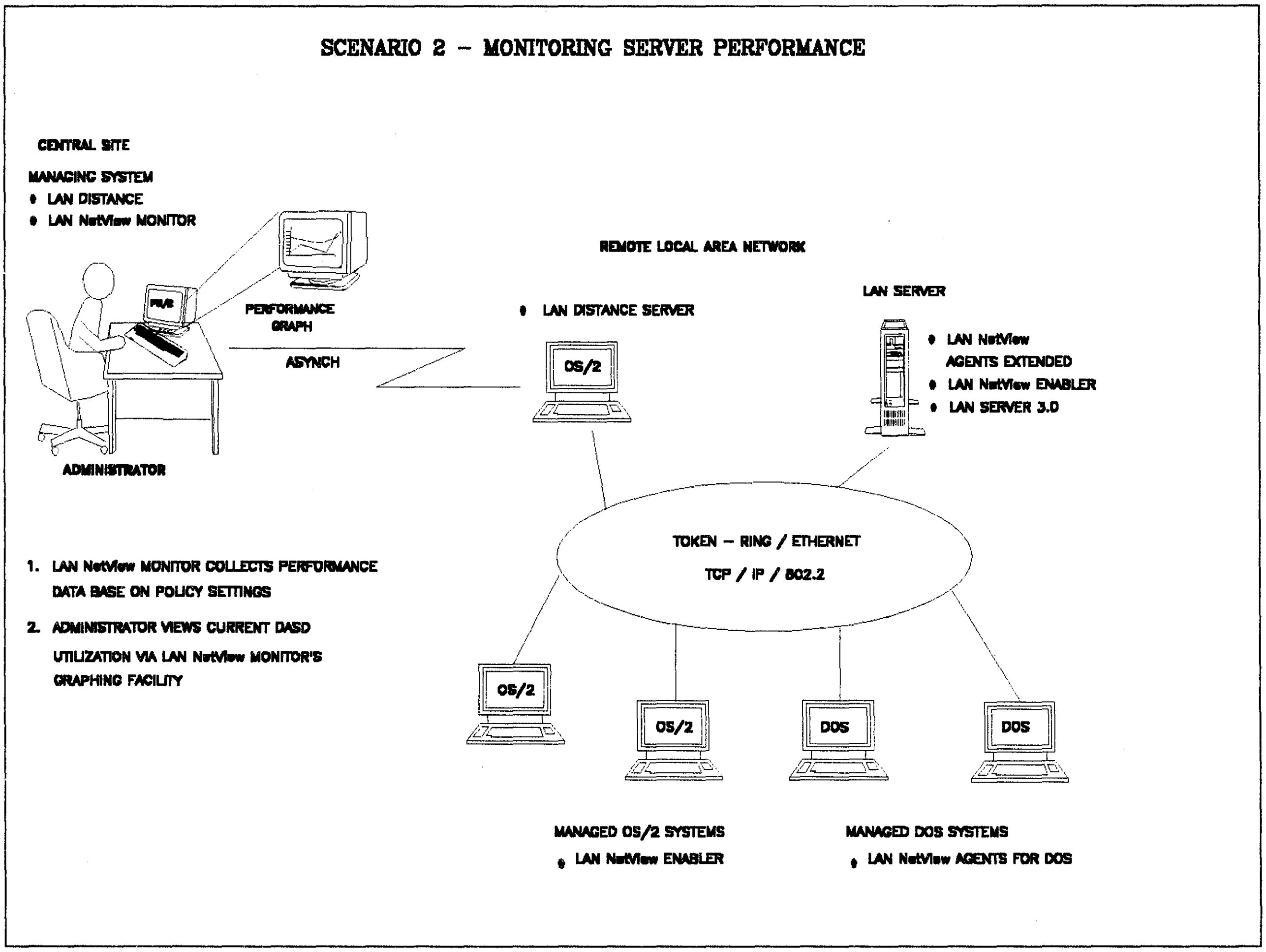


Figure 7. Scenario 2 - Monitoring Server Performance

Remote access to LAN NetView Manage 1.0 and LAN NetView Monitor 1.0 on the remote managing systems can also be achieved over the phone line connection via DCAF, using this same scenario. In this case neither LAN NetView Manage 1.0 nor LAN NetView Monitor 1.0 would be needed at the central site because the system would be used as a remote console vs. a *managing system*.

9.5.3 Scenario 3 - LAN Protocol Analysis

Protools' Network Analysis Series is a LAN performance analysis product which will be integrated with the LAN NetView platform. Foundation Manager and Cornerstone Agent are the two programs that comprise the Network Analysis Series product. It is used for performing LAN analysis functions such as protocol decoding, baselining (network health check), history logging, load emulation, and gathering performance statistics. Protocol analysis is the process of decoding protocols such as TCP/IP or NetBIOS from cryptic notation into a readable representation appropriate for reporting statistics, and/or passing to graphics applications to produce charts and graphs.

The configuration for this scenario consists of a central site using a personal workstation with OS/2 2.x, and DCAF running. The workstation is connected to a remote LAN with a mixed configuration of OS/2 and DOS systems, with LAN NetView Manage 1.0 and Protools' Foundation Manager program running on a managing system, and Protools' Cornerstone Agent program running on one of the managed OS/2 systems. In addition, there are several clients (all managed systems) with LAN NetView Enabler 1.0 installed on the OS/2 systems and LAN NetView Agents for DOS on the DOS systems. The central site is linked to the remote LAN via telephone line connection. Refer to Figure 8 on page 130.

Here the system administrator wishes to use the Network Analysis Series product remotely, to set network performance thresholds to more closely monitor traffic on this particular token-ring LAN (the product also supports Ethernet LANs as well). Before this can be done, he/she needs to ascertain what constitutes "normal" traffic. To do this, the administrator uses DCAF to gain control of the keyboard and display at the managing system at the remote site. The LAN NetView Manage 1.0 View component is used to invoke the Foundation Manager program, which is used to run a baseline check. This, over time, determines the "normal" traffic on this particular LAN. The results can then be used by the administrator to once again access the Foundation Manager program via DCAF access to LAN NetView Manage 1.0 on the remote managing system, and alarm thresholds can be set. By maintaining control of the remote managing system's keyboard and display, this type of network management, as well as additional systems management functions provided by the LAN NetView family of products, can be carried out using the remote management technique.

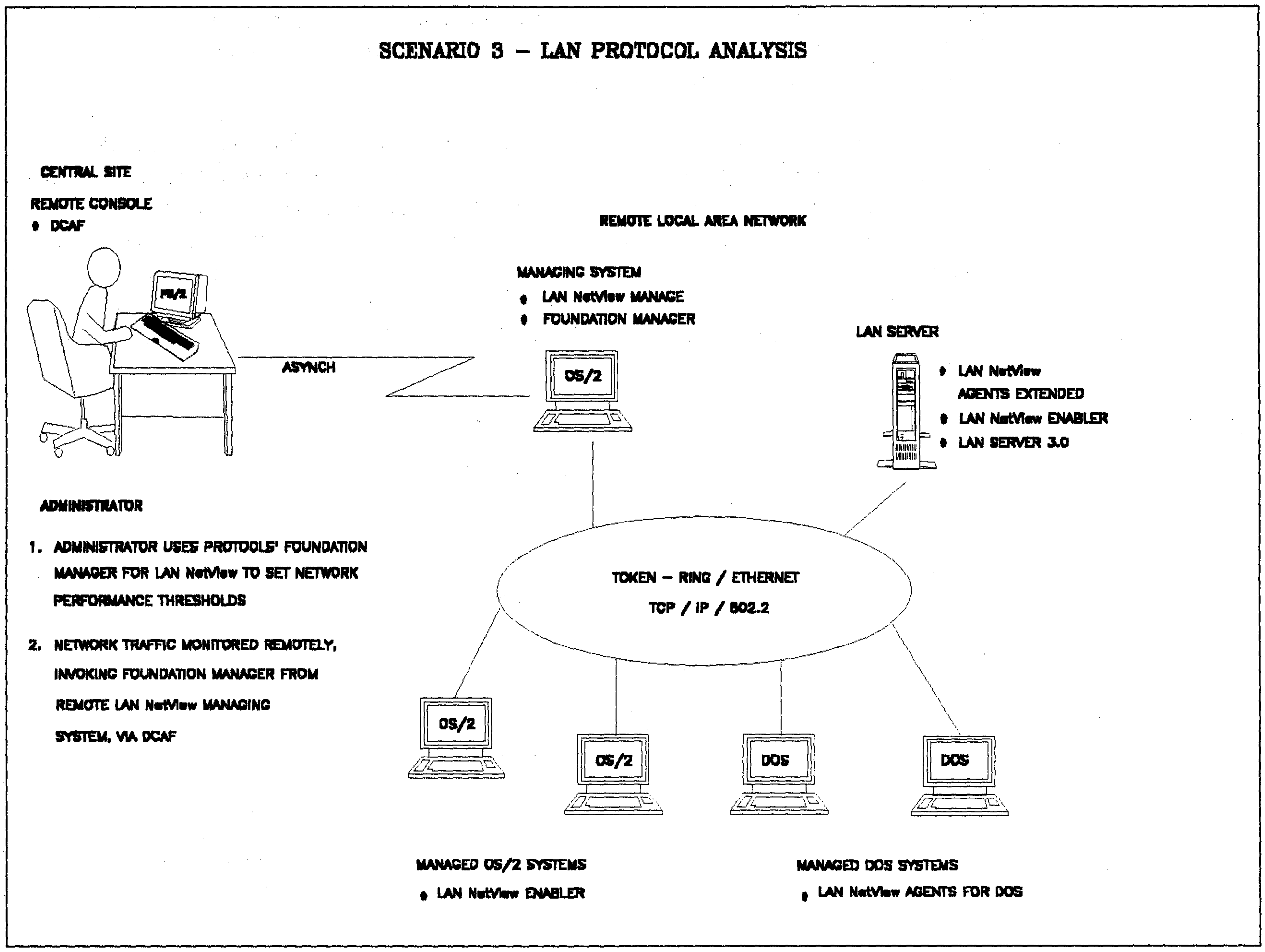


Figure 8. Scenario 3 - LAN Protocol Analysis

9.5.4 Scenario 4 - Monitoring Unattended Managing Systems

A full suite of systems management applications, including LAN NetView Monitor 1.0, LAN NetView Fix 1.0, Protools' Network Control Series, and Novell's NetWare Services Manager 1.5 for OS/2 are used in this scenario to provide a remote system administrator the means for monitoring and controlling all connected LAN workgroups.

The configuration for this scenario consists of a central site using a personal workstation with OS/2 2.x, and DCAF or LAN Distance products running. If the LAN Distance product is being used, the management applications as well as LAN NetView Manage 1.0 and agents need to run on the system. If DCAF is being used, then the system is being used as a remote console rather than a *managing system* and thus only the DCAF program is needed. In either case the workstation is connected to one or more remote LANs with mixed configurations of OS/2, DOS, and NetWare systems. One managing system at each remote site has LAN NetView Manage 1.0, LAN NetView Fix 1.0, LAN NetView Monitor 1.0, Protools' Foundation Manager for LAN NetView 1.0, and Novell's NetWare Services Manager 1.5 for OS/2 installed. The OS/2 clients have LAN NetView Enabler 1.0 running, and the OS/2 server has LAN NetView Enabler 1.0 and LAN NetView Agents Extended 1.0 installed. The NetWare server has NetWare Management Services for Netware 3.11 or 4.0 installed. The DOS systems have LAN NetView Agents for DOS 1.0 installed. If the LAN Distance product is being used, one of the systems must be configured as the LAN Distance server. The central site is linked to the remote LAN via telephone line connection. Refer to Figure 9 on page 131.

Either the LAN Distance or the DCAF product can be used for remote LAN management functions described in this scenario. System performance can be monitored on servers and clients, using LAN NetView Monitor 1.0. Network performance can be monitored on all LAN segments, using the Network Analysis Series for LAN NetView from Protools. Reporting of system and network faults is provided by the LAN NetView Fix 1.0 application. NetWare server functions can be invoked through the Novell NetWare Services Manager 1.5 for OS/2 on the LAN NetView Manage 1.0 system.

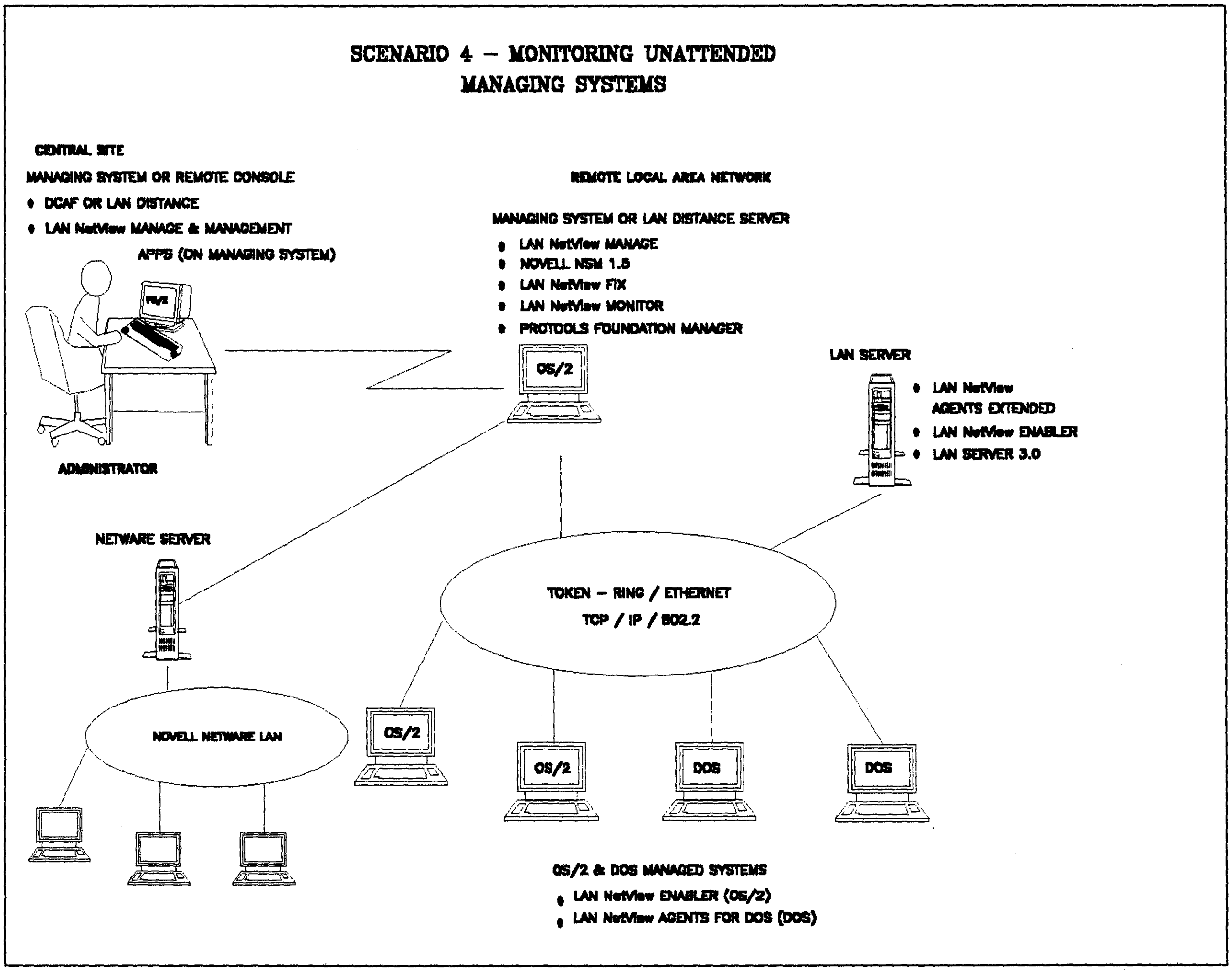


Figure 9. Scenario 4 - Monitoring Unattended Managing Systems

9.6 Summary

Remote LAN management offers LAN administrators a way to control the system and network resources efficiently and effectively in those enterprises where configurations of LANs are geographically distributed. Whether management is being performed in-house or outsourced to a service provider, the need for remote management tools can be satisfied by the LAN NetView family of products. A remote management implementation that integrates the IBM LAN

NetView platform, applications, IBM products like DCAF and LAN Distance, as well as OEM products such as the Network Analysis Series from Protools, as well as the NetWare Services Manager 1.5 from Novell, can provide a cost-effective solution for these enterprises.

Part 3. Programming Information

Chapter 10. Introduction to the IBM LAN NetView Application Programming Interface

This paper provides some basic information on the X/Open Management Protocol (XMP) and the X/Open OSI-Abstract-Data Manipulation (XOM) Application Programming Interfaces, which are used by applications to access the LAN NetView platform management services. An overview of XMP/XOM programming is provided, with some details on functions, protocol services, and sample interactions. It is intended to give application developers an insight into how applications use the API.

10.1 Introduction

This paper provides some basic information on the X/Open Management Protocol (XMP) and the X/Open OSI-Abstract-Data Manipulation (XOM) Application Programming Interfaces, which are used by applications to access the LAN NetView platform management services. An overview of XMP/XOM programming is provided, with some details on functions, protocol services, and sample interactions. It is intended to give application developers an insight into how applications use the API.

10.2 IBM LAN NetView Platform XMP/XOM Programming

The number of individual functions included within the XMP/XOM programming paradigm is relatively small, and for the most part intuitive. As with most object-oriented paradigms it is the creation/definition of the objects which is key to accomplishing the desired tasks.

From a programming perspective, all AGENT and MANAGING applications access the LAN NetView platform management services by using three function calls: mp_initialize(), mp_version(), and mp_bind().

The first of these, mp_initialize(), builds a data area or workspace to hold object definitions, and data for communication to the management services. The second function, mp_version(), identifies characteristics of the desired interface such as protocol synchronous or asynchronous response, and in some cases addressing capability.

The third function, mp_bind(), establishes a "connection" to the management services. Detailed information on parameters and returned values from these functions can be found in the LAN NetView Programmer's Reference manual.

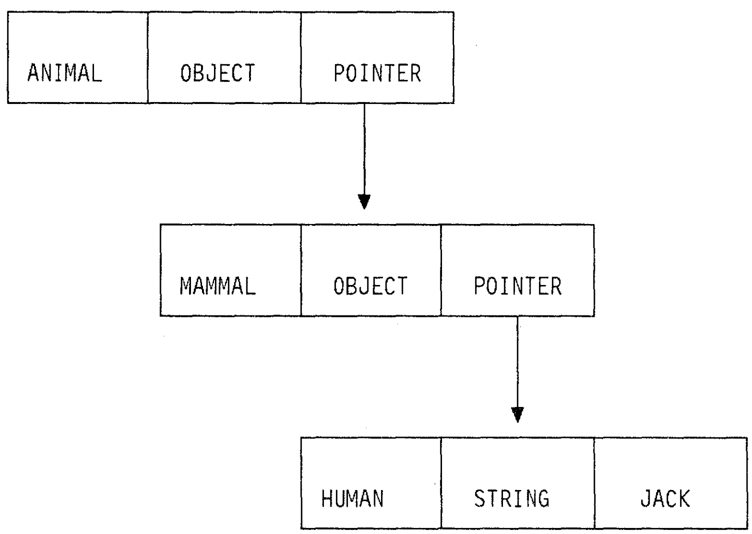
Once a connection or "bind" has been established, management requests and/or responses can be made to the management services. A managing application will typically build a sieve using mp_create(), if registration with remote agents is needed (for event-based activity), or, issue mp_get(), mp_set(), etc. function calls to perform management functions. On the managed side, an agent will typically wait for a request to arrive, using the mp_receive() function.

"Housekeeping" functions are provided to allow graceful termination or cancellation of outstanding requests by applications.

10.3 Objects

From a logical standpoint, issuing requests and acting upon them is rather simplistic. However, the key in utilizing the LAN NetView platform management services is building objects, which are complex data structures. XOM provides support for the construction and manipulation of objects. There are two broad classes of objects; public and private. Public objects may be operated upon by applications, and private objects are manipulated by the management services. An XOM function, om_get() can be used to copy a private object to a public object, and om_put() is used to do the opposite. An object within the LAN NetView platform is constructed as an array of descriptors. A descriptor is a structure consisting of 3 fields; Type, Syntax, and Value. The Name field is a descriptive term for the descriptor (that is, Object-ID), the Syntax is the equivalent of a typedef in ″C″, and the Value is the value assigned for the descriptor. The Value field takes on 3 general types of data; Integer, String (pointer to a string construct), and Object (pointer to another array of descriptors). It is this capability of nesting objects that can make the descriptors very complex.

For example:



In the above example we have a descriptor for animal, but to find the name of the animal we have to follow the object pointers two levels to human, and the name of the human. This is relatively straight-forward, and is typical of how objects are built and traversed by applications using the LAN NetView platform.

Objects used in LAN NetView platform management services must be both assembled, and disassembled for the management services to provide or get at required data values. All object classes supported by LAN NetView agents are published in the LAN NetView Managed Object Catalog. A description of the objects, and their ASN.1 definitions are included in the object catalog. It is also advisable to review the X/Open OSI-Abstract_Data Manipulation (XOM) API specification for a more complete description of descriptors, and the X/Open Management Protocol (XMP) specification for construction of descriptors.

Tables 1, 2, and 3 give a listing of many of the functions provided by XMP. Table 3 specifically lists all management functions supported by the LAN NetView product. Tables 1 and 2 identify functions needed for protocol support within XMP.

Table 4 shows the XOM function used primarily to manipulate data (objects) within the LAN NetView platform.

10.4 Arguments

Building the arguments for the management requests will take some practice. In general it is much like the example just seen, where the data you may want to provide is the name of human but the entire complex structure must be built. Each function has an associated argument object. The rules and structure for building the argument objects are described in the OSI XMP specification and the LAN NetView Programmer's Reference manual. Argument objects will vary for SNMP and CMIP protocols. Descriptions for both types of argument objects can be found in the manuals referenced.

10.4.1 Results

Results can be returned in many ways. In the simplest context, a result is returned as an object in the result parameter of the function call. This occurs when the function call is "synchronous". In other words, data is returned when the function completes, and the thread is blocked until execution completes. The data returned can be a simple object, or a list of objects, which must be traversed to obtain the results. The XOM function om_instance() can be used to identify the type of result returned, and om_get() can be used to navigate through the object layers.

Results can also be returned "asynchronously". In this case, the thread is not blocked, and control is returned to the program immediately. The function call returns an identifying tag referred to as the invoke_id. Each asynchronous function call will return a unique invoke_id (type integer). The results of asynchronous calls are retrieved by the mp_receive() function.

When mp_receive() is executed, both a result, and the invoke_id are returned. It is the responsibility of the programmer to keep track of which invoke_id went with which function call. Secondly, the programmer may determine the result type in order to efficiently parse the result returned. It is possible that multiple responses can be returned for a single asynchronous request, and multiple mp_receive() requests may be required to obtain the entire result.

The decision to use asynchronous vs. synchronous requests is essentially that of performance, space, and programming style. The equivalent of asynchronous requests using synchronous requests can be achieved if threads are started for each synchronous call. the thread will block, but the parent process/thread can continue. The reward for the extra overhead is that the result type is known, and the invoke_id does not have to be maintained.

10.5 Supporting Functions of XMP

Table 8. XMP Functions to Manage the Environment

mp_bind()	Opens a session with the OpenView communications infrastructure.
mp_error_message()	Maps an mp_status OM object into a NULL terminated string that contains an error message describing the error.
mp_initialize()	Initializes the workspace and returns a handle to the workspace.
mp_shutdown()	Discards the workspace.
mp_unbind()	Terminates a given session with the OpenView communications infrastructure.
mp_version()	Associates OM packages with the workspace (from mp_initialize()) for an application.

Table 9. XMP Functions to Support Asynchronous Activity

mp_abandon()	Abandons a pending asynchronous request.
mp_receive()	Used to obtain the argument of an asynchronous message. Responders us it to receive requests, and to receive responses to notifications. Requesters use it to receive notifications, and to obtain the results from asynchronous requests.
mp_wait()	Used to test for asynchronous data availability.

Table 10. XMP Functions to Support CMIS and SNMP Services			
CMIS Service	**SNMP Service**	**XMP Functions**	**Description (of request only)**
Action	-	mp_action_req() mp_action_rsp()	Requests the responder to perform one of the actions defined for an object.
Cancel Get	-	mp_cancel_get_req() mp_cancel_get_rsp()	Requests the responder to terminate servicing an earlier asynchronous "get" request that has not yet completed.
Create	-	mp_create_req() mp_create_rsp()	Requests the responder create an instance (object) of the specified object class
Delete	-	mp_delete_req() mp_delete_rsp()	Requests the responder to destroy a particular instance (object) of an object class.
Get	Get	mp_get_req() mp_get_rsp()	Requests the responder to supply the value(s) of one or more object attributes.
Set	Set	mp_set_req() mp_set_rsp()	Requests the responder to modify the value(s) of one or more object attributes.
Event Report	Trap	mp_event_report_req() mp_event_report_rsp()	Issues one of the notifications (events or traps) defined for an object.
	Get Next	mp_get_next_req() mp_get_next_rsp()	Requests the responder to supply the type (name) of the next SNMP variable in the object.

Table 11. Introduction to XOM Functions	
Function Name	**Description**
om_copy()	Creates an independent duplicate of a private OM object
om_copy_value()	Copy a string from one private OM object to another
om_create()	Creates a new private OM object (of a particular class)
om_decode()	Creates a private OM object that represents an encoded private OM object
om_delete()	Deletes a private or service-generated OM object
om_encode()	Create a new private OM object that encodes an existing private OM object, using the Basic Encoding Rules for ASN.1
om_get()	Creates a public copy of all or part of a private OM object
om_instance()	Tests an OM object for membership in a particular OM class
om_put()	Puts attribute values of an OM object (public or private) into a private OM object
om_read()	Reads a segment of a string in a private OM object
om_remove()	Removes (and discards) an attribute value from a private OM object (or the entire attribute itself)
om_write()	Writes a segment of a string into a private OM object

10.6 CMIP vs. SNMP

For those developing agents, both CMIP and SNMP require levels of support within the AGENT, which in turn imposes restrictions on the programmer, on the managing side. On the managing side, adherence to XMP guarantees that properly structured SNMP and CMIP requests are made to the agents. The agent process however has the task of decoding the request and performing the appropriate function and returning the appropriate result, which can be complex. The level of support and the amount of data to be returned is defined by the protocol standards.

The LAN NetView API supports both SNMP and CMIP. The key difference is the specification of the argument. The following diagram shows a sample view of the Get argument:

```
mp_get_req(session, context, get_argument, *result, *invoke_id);
                                  |
                    +-------------+
                    |             |
                    v             v
         CMIS-Get Argument     SNMP-Get-Argument

         Object Class            IP-Address
         Object Instance         Var-Id-List
         Access-Control          Access Control
         Synchronization
         Scope
```

```
Filter
Attribute-Id-List
```

In addition, the definition of what objects an AGENT supports is usually described in a MIB. MIBs use the ISO GDMO ASN.1 standard for defining objects. IBM LAN NetView Manage version 1.0 provides a compiler for compiling and placing the MIB in a metadata database for access by applications. It is the responsibility of the AGENT developer to define the MIB and provide a document similar to the LAN NetView Managed Object Catalog if it is desired to make the AGENT publicly available to managing applications.

Chapter 11. Managing OS/2, DOS and Windows Stations from LAN NetView

This paper discusses some scenarios for managing OS/2, DOS or Windows stations from LAN NetView. It focuses on using the operating system agents to implement these scenarios. It includes both an end-user perspective (using the available LAN NetView applications) and an ISV perspective (describing how to provide narrowly-focused applications that take advantage of these agents' capabilities).

11.1 Introduction

In this age of downsizing, a LAN systems administrator's job is becoming more and more complex. More networks, hardware, and applications are being added to the arena for which an administrator is responsible to manage and maintain. The operating system agents of IBM's LAN NetView family of products (OS/2, Windows, and DOS), when coupled with the LAN NetView applications, offer potential savings of time and money.

Using an object-oriented approach, based on an industry-standard open architecture, the LAN NetView product offers Distributed Systems Management capabilities that a few years ago were only dreamed of. With the open architecture not only facilitating but encouraging the development of new applications, these capabilities will continue to grow.

The objects supported by these agents have been logically defined to represent virtually all the resources available at a workstation, with each object containing multiple "attributes" that can be read (and, where appropriate, set) from remote managing workstations, using a request/response method. Furthermore, these objects are catalogued in the *IBM LAN NetView Operating Systems Agents Version 1.0 Managed-Object Catalog*, publication number S96F-8495, (more commonly referred to as the OS MOC, or the Operating Systems Managed Object Catalog). This catalog clearly defines the resources (objects and attributes) supported by each agent, leading to easy accessibility for the user or developer of a managing application).

With the LAN NetView product, such things as Asset Inventory, Workstation Memory, DASD Usage Monitoring, Software Monitoring, and problem determination are easily achieved from one or more centralized OS/2 Desktops. And with the OPEN APIs that are implemented and clearly documented, very specific applications can be developed in a timely, efficient manner.

11.2 Scoping and Filtering

Scoping and filtering are powerful management tools which are supported by all the Agents of the LAN NetView family of products. Scopes and filters, when specified in a request, are included in the request sent to an agent workstation. These techniques allow a request to be more general in nature, requiring less knowledge about the managed workstation, and resulting in less work at the managing node.

11.2.1 Scoping

Scoping is done on an object class basis, from top to bottom, based on the object-class naming hierarchy of a given system. This hierarchy may be found, in graphical form, at the beginning of the Managed Object Catalog. Scoping serves to route a given request to one or more objects within an agent workstation.

Several types of scoping are available to the managing user, and it must be remembered that all types are relative to the Base Managed Object Class (BMOC). The BMOC is the object class specified in a request as the target object of the request. Following are the types of scoping available, with a brief description of each:

- Named Numbers

  This type of scope offers the following choices of sub-types:

  - Base Object

    With this type of scope, only the Base Object specified in the request will process the request. This is actually the same as no scope.

  - First Level Only

    This type of scoped request will result in the request being processed only by the objects that are named one level below the BMOC.

  - Whole Subtree

    This type of scoped request will result in the request being processed by the BMOC plus all the objects that are named below it.

- Individual Levels

  With this type of scope, you specify the level of the object-class hierarchy (relative to the Base Managed Object Class) that you wish the request to be sent to. For example, if the request specified microsoftWindows as the BMOC, and scoping of Individual Level = 1 was chosen, the request, upon arrival at the agent workstation, would be routed to and processed by only the objects that are named 1 level below the microsoftWindows object.

- Base to Nth Level

  This type of scope results in the request being routed to the BMOC plus all objects named in the Nth levels below it.

11.2.2 Filtering

Filtering is done on an attribute basis, without regard to object classes. When an object receives a request to process, any attribute filters within the request are compared to what actually exists within the object. If the conditions of the filters are met, then the object will respond to the request, as appropriate. If the filter conditions are not met, the object will respond with an absent object, which tells the managing workstation that the object processed the request but had no response.

Several types of filters exist, and the various types may be combined to form a complex array of filtering conditions. Following are the types of scoping available, with a brief description of each:

- ITEM

This is the simplest form of filter and at least one filter ITEM must be present in a filtered request. The more complex forms of filtering are formed by using the AND, OR, and NOT filter operators on the filter ITEM.

The following choices for filter ITEM attributes are available, and are mutually exclusive:

- EQUALITY

  This filter will pass if the value of the filter ITEM is equal to the value of the attribute.

- GREATER OR EQUAL

  This filter will pass if the value of the filter ITEM is greater than or equal to the value of the attribute.

- LESS OR EQUAL

  This filter will pass if the value of the filter ITEM is less than or equal to the value of the attribute.

- PRESENT

  This filter will pass if the attribute specified in the filter ITEM is defined within the object class.

• AND

  Use this filter type when the filter condition is that two filter ITEMs are true.

• OR

  This filter type is used when the filter condition is that at least one of two filter ITEMs are true.

• NOT

  This filter type is used when the filter condition is that the following filter ITEM is not true.

11.3 An Example

It's 10:00 on a Monday morning, and you're back at your desk after a meeting where you were asked to identify how many workstations in your shop will need a memory upgrade (say, for example, to 12 Megabytes) to support some new applications that are rolling out next month. As the System Administrator, you know there are around 100 machines that will need to be checked. Luckily, a week earlier, you saw to it that LAN NetView Version 1.0 was installed across your network, and you just happen to have one of the managing workstations at your fingertips. You've only played with this system a little, you have a few doubts, but still you're convinced this task can be achieved easily.

Since your network is composed of DOS, OS/2, and DOS+Windows workstations, you quickly determine that all you need to do is bring up the View screen, click on the Request Manager for each workstation in question, then issue a get request to each of those workstations.

The attribute requested will be the Total Physical Memory. For each of the different operating systems (DOS and OS/2), this attribute is defined in a different object class: dosOperatingSystem object on DOS systems and operatingSystem2 object on OS/2 systems. Therefore, to make the request as generic as possible, you'll want to use scoping.

The System Object, which is at the top of the naming hierarchy and is modelled by each of the OS Agents, will be the Base Managed Object Class (BMOC) for the request. This object class, defined in the DMI standard, does not contain any attributes that are relevant to the task at hand, but its presence at each of the OS Agent workstations provides an easy way to issue scoped requests to the various workstations without needing to know whether the operating system is DOS or OS/2.

The scope type could be whole subtree, which ensures that all of the objects under the System Object will receive the request. However, after referring to the Naming Hierarchy in the GDMO, you see that "first-level only" scoping will also work, since the object classes where the Total Physical Memory attribute is defined are all named one level below the System Object. Since the first-level only case will require less processing at each agent workstation, you decide to use this type of scope, for performance reasons.

The request is actually quite simple. In Managed Object Catalog (MOC) terms, it reads like this: IF (((the dosOperatingSystem object exists) OR (the operatingSytem2 object exists)) AND (totPhysMemory is less than 12 meg)) then return the value of the totPhysMemory attribute).

Using the graphical user interface of the Request Manager, by clicking a mouse button, you have built this request and caused it to be sent to the first of the workstations in question. In a few seconds, the response has arrived back at your machine. If the response contained the totPhysMemory attribute value, you know that workstation has less than 12 MB. If the response only contained an "absent object", you know that the workstation already has at least 12 MB installed (that is, the filter failed). Responses that contain "invalid object instance" will indicate machines that are not running. At the Agent workstation, business continued as normal, with the user unaware that this request was processed in the background.

You now only need to repeat this request to each of the other machines and compile your report from the data received. Only the machines with less than 12 MB of memory will respond with an attribute value, and the amount of memory that needs to be added to each can be easily deduced because each response contains the amount of total physical memory that is currently installed.

While this procedure will accomplish your goal, you quickly realize that a special application to broadcast a given request to a pre-defined list of workstations would be easily achievable and very useful. You make a mental note to request just such a tool from your programming support team.

In the example above, scoping and filtering techniques were used to identify all the workstations in the network with less than 12 megabytes of memory installed. Using the machine location information which is configured when the OS Agents are installed, filters could easily be added to this example to retrieve the same information for only the workstations located on the fourth floor of building X and the fifth floor of building Y.

For those users who like to write programs, a narrowly-focused application could easily be written to do all of that request, then compile the data into a report.

11.4 Summary

In summary, the Systems Management possibilities with the IBM LAN NetView family of products are nearly endless. With the combination of the LAN NetView family of management applications, offerings being developed by ISVs, the open API architecture available to anyone wishing to develop specific management applications, and with IBM's commitment to future enhancements of this product, the LAN NetView product will prove to be an industry standard for the 90's and beyond.

Chapter 12. IBM LAN NetView: Agents and Objects Overview

This paper discusses the IBM LAN NetView Agents and LAN NetView Agents Extended products and the Management Information Base (MIB) they represent.

This paper is intended for IBM marketing representatives, IBM system engineers, customers and software developers who desire a general understanding of the systems management capabilities provided by the IBM LAN NetView product. It focuses on the LAN NetView agent products and the potential for using them when developing systems management applications.

12.1 Introduction

The IBM LAN NetView family of products provides a framework and applications to implement OS/2-based distributed systems management solutions. The framework utilizes industry standard interfaces and protocols that allow an OS/2 system to manage heterogeneous systems in a LAN environment. An OS/2 system may also be managed by other systems that conform to the same standards.

The LAN NetView family of products is based on systems management standards such as those developed by ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) as part of their work on Open Systems Interconnection (OSI).

The primary purpose of this paper is to provide the reader with a look at the various agents that are shipped as part of the LAN NetView family of products and the Management Information Base (MIB) they represent. These agents provide the interfaces to managed systems that are required by management applications.

The LAN NetView agents are a key component of the LAN NetView family of products. They provide systems management application developers with access to a wide variety of objects associated with OS/2, its major subsystems (LAN Server V3.0, Database Manager and Communications Manager), DOS and Microsoft Windows. In addition these agents provide information regarding the hardware on which they are running.

Agents are systems management applications which perform operations on managed objects at the request of managing applications and emit notifications on behalf of managed objects.

IBM LAN NetView systems management applications will take advantage of these agents. However, and maybe more importantly, these agents may be utilized by systems management applications written by other vendors or customers. The MIB defined by these agents represents a rich set of object classes which can be used by applications that span all functional areas (or disciplines) of systems management.

This paper will briefly introduce the LAN NetView concepts and products, and then provide a more in-depth look at the MIB defined by the various agents and the possibilities it presents for systems management applications.

12.2 The LAN NetView Framework

The LAN NetView family of products is based on a series of industry standards as mentioned previously. This family of products provides systems management applications and a framework. This framework provides an enabling platform for IBM applications as well as applications written by vendors or customers. The following sections will provide a brief overview of the concepts behind the LAN NetView framework.

12.2.1 The Object-Oriented Model

IBM LAN NetView is based on a variety of industry standards. These standards include those developed by ISO for the management of OSI-based networks. The ISO standards in this area define protocols to be used between managing and managed systems. In addition to protocols, the ISO standards provide a common method for the definition of managed resources. These standards, are often referred to as GDMO (Guidelines for the Definition of Managed Objects), allow agents representing managed resources to be developed independently of the applications which will manage those resources.

When using the OSI model for systems management, managed resources are represented as objects.

In the LAN NetView framework environment, a managed object is an abstraction that models some physical resource (such as a workstation) or logical resource (such as a file system).

Those aspects of resources that are related to the management of the resource are accessible through the managed-object abstraction.

A managed object is defined by:

- Attributes

  The attributes represent information about the object.

  Operations may be performed on specific attributes in order to GET or SET their values.

- Actions

  Actions represent functions that an object can perform.

- Notifications

  A notification is an unsolicited message that an object emits regarding a change in state or status or to notify the receiver of some other significant event.

A set of object classes and their associated attributes, actions and notifications make up a Management Information Base (MIB). Management applications may then be designed to utilize the MIB in performing management functions.

12.3 Managers, Agents and Managed Object Interaction

A distributed systems management environment is achieved through distinct entities called managers and agents. This environment is similar to a client-server model.

An agent is the part of a distributed management program that supervises one or more managed objects. The agent receives requests for operations to be performed on managed objects or requests for objects to perform certain actions. The agent is responsible for passing these requests to a resource manager. A resource manager provides the interfaces required to carry out the specific request.

For example, in an OS/2 environment, the LAN Server, Communications Manager, Database manager or even the operating system itself can be considered a resource manager.

The agent is also responsible for emitting notifications (events/traps) when it detects special conditions in the managed object.

A manager is the part of a distributed management program that issues requests for actions and receives notifications. A manager uses the services of one or more agents. Managers do not manage resources directly, rather they issue requests to objects which are represented by agents.

Figure 10 represents the interactions between managers, agents and managed resources.

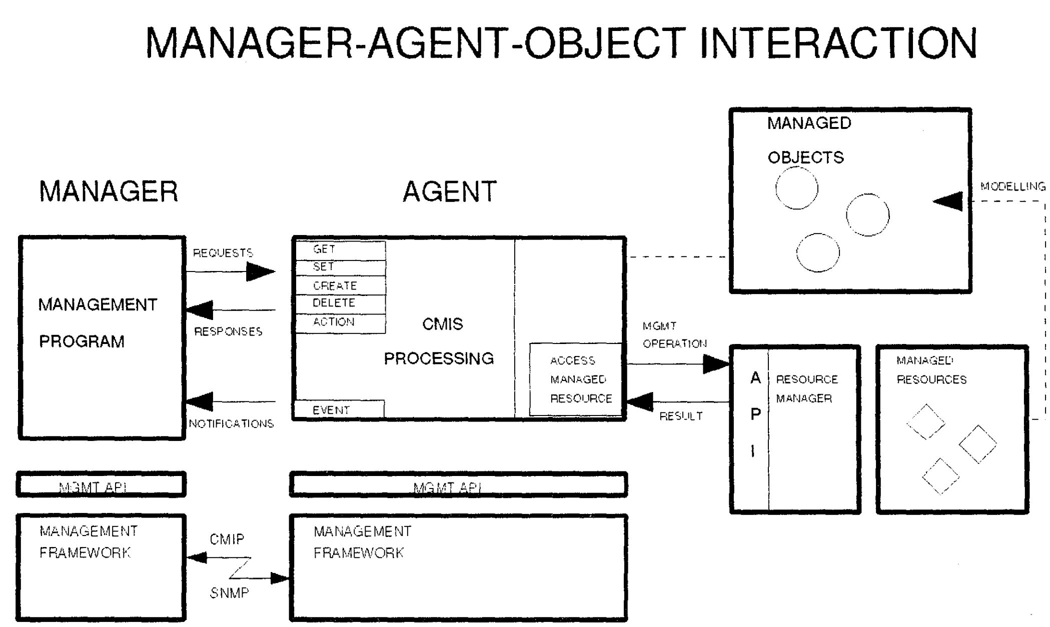


Figure 10. Systems Management Interactions

Each subsystem which controls a set of resources is called a resource manager. Examples include:

- the Operating System
- the Communications Manager
- the Database Manager
- the LAN Server

An agent will perform management operations on resources through API's provided by the resource manager.

The interfaces used between an agent and the resources it represents are not subject to standardization. An agent is free to use whatever interfaces are available in the system in order to carry out operations or actions. Even though the framework is object-oriented, the actual interaction between the agent and the resource manager may use non-object oriented interfaces. This interaction is completely hidden from the managing system.

12.4 Standards Used by LAN NetView

The LAN NetView architecture is based on and utilizes a variety of industry standards, including:

- 8824 (X.208): OSI - Specification of Abstract Syntax Notation One (ASN.1)
- 8825: OSI - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)
- ISO/IEC 9595 (X.710): Common Management Information - Service Definition
- ISO/IEC 9596-1 (X.711): Common Management Information - Protocol Specification
- ISO/IEC 10165-1 (X.720): Structure of Management Information - Management Information Model
- ISO/IEC 10165-2 (X.721): Structure of Management Information - Definition of Management Information
- ISO/IEC 10165-4 (X.722): Structure of Management Information - Guidelines for the Definition of Managed Objects
- RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets
- RFC 1157: A Simple Network Management Protocol (SNMP)
- RFC 1213: Management Information Base for Network Management of TCP/IP-based Internets: MIB-II

The LAN NetView architecture currently supports two different sets of protocols and services to be used between managing and managed systems:

- CMIP, the Common Management Information Protocol was designed for managing OSI networks. The services defined for CMIP are known as the Common Management Information Services, or CMIS.
- SNMP, the Simple Network Management Protocol was designed for managing TCP/IP networks and devices.

The LAN NetView framework provides three key APIs for building applications:

- The X/Open OSI-Abstract-Data Manipulation (XOM) API
- The X/Open Management Protocol (XMP) API

- The Graphic User Interface (GUI) API

The XOM API is used to manipulate the data structures associated with objects.

The XMP API is used for standards-based process to process communications between a managing system and a managed system.

The GUI API is used by applications on the managing system to provide a consistent user interface to all management applications using the LAN NetView framework, and to enhance the capability for seamless navigation between systems management applications for the user.

12.5 The LAN NetView Family of Products

The LAN NetView family of products includes support for both managing and managed systems. The environment is depicted in Figure 11.

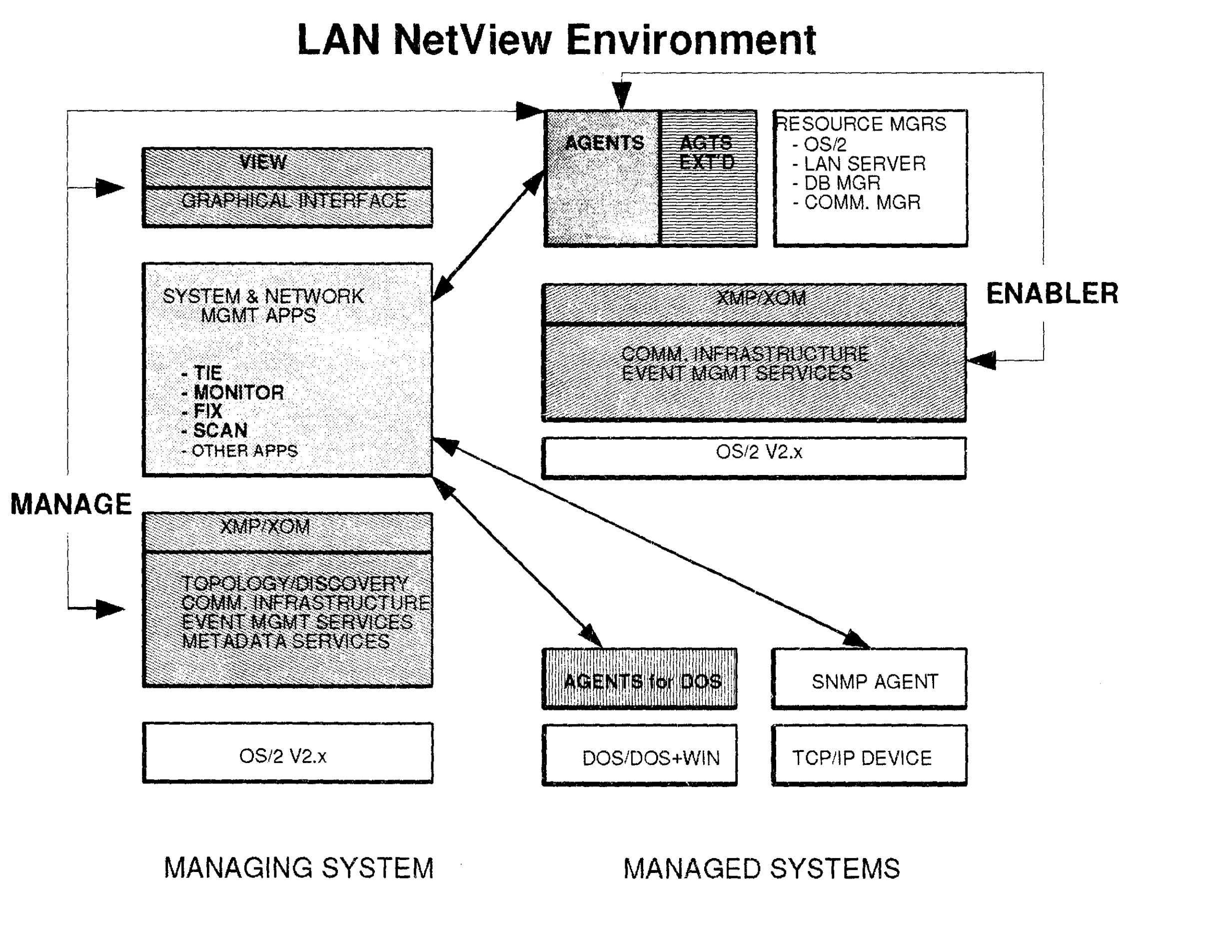


Figure 11. LAN NetView Environment

The following sections will briefly describe the products that comprise the LAN NetView products and the roles they play.

12.5.1 LAN NetView Manage

The LAN NetView Manage product (hereafter referred to as Manage) provides the core functions required by a managing system. These include a communications infrastructure, event management, metadata and topology/discovery services. Manage provides the industry standard X/Open Management API's (XMP and XOM) for the development of management applications. By utilizing the XMP API's, applications using the LAN NetView Manage framework can be written to use either SNMP or CMOT/CMOL protocols.

In addition, Manage includes the View component which provides a graphical interface to the LAN NetView framework. Developers may use the View programming interfaces to deliver a consistent look and feel to their management applications.

The View component will automatically provide for easy navigation through a series of applications. It also provides services to allow for the displaying of management data in progressive layers of detail.

Manage will also provide the OS/2 and LAN Requester agents. These agents are normally installed on a managed system. In some cases, a managing system may itself be managed by another managing system. Also, a managing system may include itself as a managed system when gathering management data. In either of these cases, these agents will may also be installed along with Manage on a managing system.

Two application level functions are also provided with Manage: the Request Manager and the Remote Command Line Interface (RCLI). The Request Manager allows the system administrator to access function in IBM agents and other CMIP and SNMP agents. The Request Manager allows the user to query attribute values of objects represented by agents as well as to set selected attribute values. This utility can be used to provide management functions that are not available by using any of the available applications.

The RCLI allows the LAN administrator to issue commands from a managing workstation to be executed on a managed workstation. This function will be useful for such things as starting and stopping OS/2 applications, changing configuration values in OS/2 applications and querying the status of an OS/2 system.

12.5.2 LAN NetView Enabler

The LAN NetView Enabler product provides the managed system platform on OS/2 V2.0x-based systems. It also provides the XMP/XOM programming interfaces for the development of management agents which will interact with applications on the managing system.

LAN NetView Enabler 1.0 also provides the OS/2 and LAN Requester agents that will allow a managing system to manage those resources controlled by the base operating system or the LAN Requester.

12.5.3 LAN NetView Agents for DOS

This product provides the agents required for IBM DOS V5.0/6.1, Microsoft DOS V5.0/6.0, and Microsoft Windows V3.0/3.1. IBM's managing applications will require that these agents are installed on any DOS/Windows systems that are to be managed.

12.5.4 LAN NetView Agents Extended

The LAN NetView Agents Extended product provides the agent support for the LAN Server V3.0, Database Manager and Communications Manager subsystems.

12.5.5 LAN NetView Applications

The following management applications have also been announced by IBM. These applications will run on a managing system (requiring LAN NetView Manage 1.0) and will utilize the agents on the managed systems.

LAN NetView Fix - is a general purpose event handling application that enables software products to automate their problem determination procedures in a manner that can be tailored and extended by customers. It receives and processes both CMIP events and SNMP traps that are emitted from managed machines on the network. Users specify the actions that LAN NetView Fix 1.0 is to take based on the events received. Both LAN NetView Fix 1.0-supplied and user-written actions can be specified.

LAN NetView Monitor - provides automated performance management through the use of user-defined policies that specify the resource data to be collected, collection schedules, threshold levels and actions, and data transfer times.

Data may be collected and stored in a relational database. A graphical display of data collected can be presented as it is collected or from data previously stored in the database.

LAN NetView Tie - provides a mechanism for the filtering and transmission of notifications emitted on the LAN to a NetView host. The LAN NetView Tie program can register to receive both OSI-alarm and non-alarm notifications. The program converts OSI alarm notifications to SNA alerts or resolutions. Non-alarm CMIP events are wrapped in a SNA/MS major vector along with other related information and sent to the host. A receiving application must be available to unwrap and parse this information.

The LAN NetView Tie product can be configured by an administrator or through NetView RUNCMD's at a host-based NetView console.

LAN NetView Scan - is currently available through a Beta program. The announcement of this product's availability will be determined by the experience of the Beta participants and the feedback they provide to IBM on the LAN NetView Scan product's function and usability.

The LAN NetView Scan product provides function for configuration and inventory management of LAN-attached workstations running OS/2, DOS and DOS with Windows.

The LAN NetView Scan product can collect workstation Vital Product Data as well as monitor and collect workstation files. The information and/or files that are gathered are stored in a centrally-located SQL database.

The LAN NetView Scan product also provides a provides a command scheduler. At regularly scheduled times, it will run programs or commands at selected OS/2 workstations to perform such tasks as software inventory, system backup, virus checking, system diagnostics and report generation.

12.6 LAN NetView Agents

In 12.2, "The LAN NetView Framework" on page 150 we discussed the role of agents in a distributed systems management environment and in12.5, "The LAN NetView Family of Products" on page 153 we described the packaging of the LAN NetView agents.

This chapter will provide additional detail and introduce a few new concepts that are key to understanding the agents and the MIB.

12.6.1 Resource Manager Agents

The various LAN NetView agents define objects for the following resource managers:

- the operating system
  - OS/2
  - DOS V5.0/6.0/6.1
  - DOS V5.0/6.0 w/ Microsoft Windows V3.0/3.1
- LAN Server V3.0
- Communications Manager
- Database Manager

In addition there is a 'system agent' which assists with topology discovery and provides the facilities required for the Remote Command Line Interface (RCLI) provided by the Manage product.

The agents will define objects and their associated attributes, actions and notifications related to the resource. A summary of these agents and objects is provided in the section "LAN NetView MIB Summary".

It is important to note that the DOS and OS/2 operating system agents were designed with as much commonality as possible. This allows management applications to be written to manage any of the operating system environments with a minimum of operating system specific code. Many of the same object classes are defined under the different operating system agents.

For example, there are several object classes related to hardware which are defined by the operating system agents. Therefore, it is possible to reference the same object class (such as Fixed Disk Drive) from a management application that is managing both DOS and OS/2 systems.

12.6.1.1 Resource Manager Objects

The majority of the objects defined by the agents relate to specific physical or logical resources and the attributes associated with those objects can be mapped to specific information about those resources. The managing application addresses these resources as objects. The agent uses whatever API's are made available by the resource manager to access the resources and the associated information.

Accessing the attributes is usually accomplished by a managing application issuing a GET or SET request to the appropriate object. This returns or modifies the specific attribute. If multiple attributes are involved, then this could require multiple GET or SET operations.

This is fine for most management applications, but there are some cases where a slightly different mode of operation is required. In these cases we will define special objects called Management Support Objects, which is the subject of the next section.

12.6.1.2 Management Support Objects (Monitor-Scanner)

In support of the LAN NetView Monitor application or other applications wishing to gain access to system performance data, the agents have defined special objects which we will call management support objects. The methods associated with these objects utilize performance instrumentation provided by OS/2 V2.x. This instrumentation provides access to low level counters, timers and control blocks necessary for performance management.

There are several reasons why performance data may need to be handled differently than other attributes:

- There must be synchronization of the access to different attributes.

  A series of GET functions to take a snapshot of multiple attributes would not be adequate as each would return attribute values from a different point in time.

- There is a requirement to monitor or scan certain counters/timers.

  This provides the capability to generate a notification when certain thresholds are reached.

- There is a requirement to log or capture certain data over time and at specific intervals.

- Applications may only require a summarization of the data collected. This in turn would also reduce network traffic.

OS/2's performance instrumentation provides the system level functions required to achieve each of the above objectives. However, implementing this through the standard GET/SET interfaces to the resource manager objects would be difficult if not impossible.

Therefore a special monitor-scanner object class is provided to provide the necessary interfaces for a managing application. The monitor-scanner object will interface directly with OS/2 to provide the functions required.

The performance information can be mapped to attributes associated with resource manager objects. In fact, the MIBs associated with the OS/2 and LAN Server agents will define the attributes associated with the monitor-scanner object. However, a managing application would not issue a GET or a SET to the resource manager object agent to retrieve or modify these attributes.

Instead, the managing application will CREATE an instance of the monitor-scanner object. In creating this instance, it will define such parameters as the specific attributes to monitor, the granularity period (how often it scans the information), thresholds, schedules and more.

Once the monitor-scanner object is instantiated, the performance data will be gathered/monitored by the agent code. Notifications will be generated when thresholds are exceeded. If a managing application wishes to retrieve the monitored information, it then issues a specific ACTION which will return the logged data.

There are other management support objects related to the monitor-scanner and the log files that it generates. The information related to these objects and how to access the information they represent will be supplied in the LAN NetView product MIB documentation.

In the section entitled "LAN NetView MIB Summary" we provide a summary of the LAN NetView MIBs. We do not specifically cover the monitor-scanner object. The information it represents is listed as attributes of the objects defined by the OS/2 and LAN Server agents.

We handled the attributes in this way in order to more easily summarize the available information. The actual process for accessing the individual attributes (either through the resource manager object or the monitor scanner object) is beyond the scope of this paper.

12.7 Solutions Based on LAN NetView Agents

This chapter will provide the reader with a look at the MIB that is defined and implemented through the LAN NetView agents.

This rich set of object classes will allow applications to be written that take advantage of the IBM-supplied agents. This will relieve application developers of the requirement to write their own agents and will increase consistency across applications developed independently.

This chapter should not be considered to be a comprehensive look at all of the available attributes, actions and notifications defined by the LAN NetView agents' MIB, but rather a sampling that represents a subset of those provided.

We will present this information by looking at five primary systems management functional areas:

- Operations Management
- Configuration Management
- Problem Management
- Performance Management
- Inventory Management

Though most of these areas are addressed in part by the IBM LAN NetView products, customers may desire additional management applications which meet their specific needs.

The primary objective of IBM's agent design team was to provide support for the monitoring and control of the operating systems and subsystems.

The secondary objective was to monitor and control specific resources owned by the operating systems and subsystems.

Each of these objectives were addressed in light of the five functional areas listed above.

12.7.1 Operations Management

Operations Management functions address the capability to monitor and alter the operational state of the systems and subsystems within a network. The primary functions include:

- Monitoring the status of resources
- Querying/changing operational characteristics of resources
- Performing actions on resources

Examples of Operations Management Tasks

The following are examples of the types of operational management that would be desired in a distributed OS/2 environment:

- Monitoring status of critical systems (LAN Servers, Database Servers, etc.)
- Monitoring status of peripheral devices such as printers
- Monitoring status of critical processes or threads
- Holding or releasing print queues
- System shutdown/restart

12.7.1.1 Monitoring the Status of Resources

The resources within the network that require monitoring can include both hardware and software.

When monitoring the status of resources within the network, it is desirable to be able to receive changes in status through unsolicited notifications. It is equally desirable to be able to explicitly query the status of a particular resource. Both of these capabilities are supported by the LAN NetView agents.

The ISO/IEC 10165-2 (X.721): Structure of Management Information - Definition of Management Information specification defines the state attributes listed below. The LAN NetView agents will use these states as applicable.

- **Administrative State**
  - the resource is prohibited (locked)
  - permitted for existing users (shutting down)
  - permitted to perform services for its users (unlocked).
- **Operational State**
  - the resource is totally inoperable (disabled),
  - partially or fully operable (enabled)
- **Availability State**
  - test
  - failed
  - power off
  - off line
  - off duty
  - dependency
  - degraded
  - not installed
  - log full

One or more of the above states may be applicable to a particular resource at any moment in time.

- **Usage State**
  - the resource is currently not in use (idle)
  - in use and has enough capacity to accommodate additional users (active)
  - in use but does not have enough capacity to accommodate additional users (busy).

There are additional resource specific states that have been defined by the LAN NetView agents.

The following is a list of some of the objects represented by the LAN NetView agents that will provide status information:

- Hardware Objects
  - Machine
  - Display
  - CPU
  - Logical ports
  - Physical Storage Devices
- Operating System Environments
  - OS/2
  - DOS
  - Windows
- OS/2 Programming Objects
  - Processes
  - Threads
- Printing Objects
  - Printers
  - Queues
  - Jobs
- Subsystem Objects
  - LAN Server Objects
    - LAN Server
    - LAN Requester
  - Communications Manager Objects
    - Configuration File
    - Conversation
    - Communications Manager
    - Transmission Groups
    - APPN Node
    - Logical Link
    - Transaction Program
    - SNA Session
    - Physical Port
  - Database Manager Objects
    - Database Manager
    - Database
    - Database Gateway

12.7.1.2 Querying/Changing Operational Characteristics of Resources

In addition to querying the operational status of a resource, an operations management program can issue a GET operation to retrieve object attributes which contain information about the operational characteristics of managed resources.

The following list provides examples of some of the objects and attributes that provide operational information. Many attributes can also be altered through a SET command issued to the agent.

- Logical Volumes : Volume Size, Allocation information, Utilization.
- Monitored Files : Content (CRC value), Size, Last modification date/time.
- Printer : Current print job, Printer status
- Print Queue : List of printers, Priority, Queue Status, Number of jobs in the queue
- Spooled Job : Printer name, Position in Queue, Name of user
- LAN Server : Service Statistics, Current Status
- Communications Manager : Status and information for APPN Nodes, Logical Links, SNA Sessions, Transaction Programs
- Database Manager : Database State (Consistent, Requires back-up, Roll-forward in progress)

12.7.1.3 Performing Actions on Resources

An operational management application should also be able to initiate actions to be performed by an agent on an object.

The following list provides examples of some of the actions that are defined and supported by the LAN NetView agents:

- Shutdown/Restart the operating system
- Hold/Release a queue or a job in a queue
- Pause/continue printing
- Pause/continue LAN Server services
- Clear LAN Server statistics
- Activate/deactivate Communications Manager, APPN node
- Activate/deactivate adapter
- Deactivate logical link
- Deactivate LU 6.2 session
- Activate/deactivate Database Manager
- Activate/deactivate a database
- Create/Drop a database.

It should also be noted that the Remote Command Line Interface in conjunction with the system agent provides the capability to execute commands on the managed workstation.

12.7.1.4 Operations Management Summary

The LAN NetView agents:

- provide access to a wide array of status information
- allow for operational characteristics to be queried and altered
- permit operational actions to be performed

on a managed system's hardware, operating system and supported subsystems.

12.7.2 Configuration Management

Configuration management involves the capability to determine, alter and track the configuration of systems within the network. These tasks can be summarized as:

- Retrieving the configuration of selected systems
- Changing configuration parameters on selected systems
- Receiving notifications of configuration changes to selected systems

The LAN NetView agents provide access to both hardware and software configuration information.

Examples of Configuration Management Tasks

The following are examples of the types of configuration management tasks that would be desired in a distributed environment:

- Query physical configuration of devices in the network
- Query/change system configuration parameters (for example, CONFIG.SYS)
- Query/change IBM LAN Server configuration parameters (IBMLAN.INI)
- Query Communications Manager configuration files
- Query/change Database Manager configuration parameters
- Receive notifications of changes to configurations on monitored systems

12.7.2.1 Retrieving Configuration Information

The following lists summarizes some of the configuration information which can be retrieved from a managed system running the LAN NetView agents.

- Operating System : Version/Level
- Memory : installed memory, memory allocation (OS/2), expanded extended memory allocation (DOS), memory configuration virtual memory (WINDOWS)
- Current system : Boot drive, Time/Date
- Code page, Country code
- System parameters (CONFIG.SYS)
- LAN Server : Version/Level, Initialization parameters (IBMLAN.INI), Run-time parameters
- Communications Manager : Version/Level, List of Configuration Files, Active Configuration File
- Database Manager : Version/Level, Configuration Information

12.7.2.2 Changing the Configuration

The LAN NetView agents allow a management program to alter various configuration parameters through a SET operation. The following list summarizes these capabilities:

- Various parameters specified in the CONFIG.SYS file.
- LAN Server : IBMLAN.INI and run-time.

  All start-up (IBMLAN.INI) parameters may be set. Selected run-time parameters may also be set/modified by a management program.

- Database Manager : Catalog/Uncatalog Databases, Create/Delete Database directory information, Create/Delete Database Connection Services Information, Create/Delete Node Directory information
- Actions exist for resetting database and database manager configuration values to defaults.

12.7.2.3 Tracking Changes to Configurations

The agents will generate notifications for the creation or deletion of some resources and will generate notifications for changes to certain attributes such as configuration parameters.

In addition, there is a 'Monitored File' object class that will allow an application to monitor critical files (such as configuration files) and be notified if they are changed.

12.7.2.4 Configuration Management Summary

The LAN NetView MIB provides the prerequisite object class definitions to develop a powerful configuration management application which can access most configuration parameters within a managed system. The capability to be alerted to changes to 'monitored files' provides a key function for identifying changes to configuration files on key systems within the network.

12.7.3 Problem Management

Problem management involves the monitoring of resources, analyzing notifications emitted from resources and performing actions to correct, avoid or circumvent error conditions.

Examples of Problem Management Tasks

The following are examples of the types of problem management tasks that may be performed in a distributed environment:

- Monitor the 'heartbeat' of agents for critical systems/subsystems
- Receiving 'alarms' generated by the LAN Server/ Requester
- Receiving 'alarms' associated with the Database Manager
- Receiving 'alarms' associated with the Communications Manager
- Perform actions to correct, avoid or circumvent error conditions

12.7.3.1 Monitoring Resources

Agents will send 'heartbeat' notifications at startup and just before normal termination. Optionally, they may also send periodic heartbeats which allows a managing system to recognize when a system or agent has had an abnormal termination.

Agents will send notifications as objects are created or deleted. For instance, a notification will be generated when the Communications Manager becomes active on a node.

12.7.3.2 Analyzing Received Notifications

It is the responsibility of the managing application to analyze the notifications it receives from agents in order to help determine the cause of failure.

Examples of the type of notifications that are generated by the agents on behalf of the managed resources are listed below:

- LAN Server
  - Quality of service alarms : Network I/O error threshold reached, disk drive nearing capacity, audit log full, etc.
  - Equipment alarms : Fault Tolerance system fixed a bad sector, Fault Tolerance system detected a difference between the contents of the primary and the secondary partitions of a mirrored fixed-disk drive, etc.
  - Environmental Alarms : LAN Server has detected multiple failed password-entry, multiple unauthorized resource-access attempts, etc..
- LAN Requester
  - Quality of service alarm : Error log has reached its maximum size, redirector has reached the configured threshold for a specified resource
  - Processing alarm : Internal processing error, resources not available, etc.

  Detailed information about each error condition will be transmitted in the notification.
- Communications Manager
  - APPN node : Insufficient storage for intermediary session setup (only by APPN network node), SNA protocol error, etc.
  - LAN adapter used for SNA : open failure detected by token-ring lobe , CSMACD bus inoperative

- SDLC adapter used for SNA : Link error due to the remote link station address, link error due to bad line, etc.

  Error conditions that normally generate SNA alerts in the Communications Manager will cause the CM agent to generate a notification to a managing system.

  The notification carried information about the event (probable cause, specific problem, severity, proposed repair actions), and the problem data (product IDs, alert type, failures cause).
- Database Manager
  - Processing Error Alarm : Internal processing errors
  - State Changes
  - Creation/Deletion of Objects

  Notifications will include the SQL return code if applicable.

12.7.3.3 Corrective Actions

Many of the capabilities discussed under other functional areas such as operations management and configuration management apply equally as well within problem management. Once a problem has been detected and the cause determined, operations or actions are typically performed to carry out the problem resolution.

For example, a configuration change might be required with a system restart to follow. These capabilities are addressed in the sections relating to configuration and operations management.

The RCLI facility allows the administrator to execute commands on the managed system in order to correct identified problems.

12.7.3.4 Problem Management Summary

The LAN NetView agents provide a rich set of resource monitoring and notification capabilities across all resources. The capabilities exist to perform corrective actions once the proper resolution has been determined.

12.7.4 Performance Management

Performance management involves the monitoring of resources to identify potential or real performance problems, isolate the causes and to correct the situations through load balancing and reconfiguration.

Examples of Performance Management Tasks

The following are examples of the types of performance management tasks that may be performed in a distributed environment:

- Tracking the current resource utilization on critical systems (LAN servers, DB Servers)
  - CPU
  - Memory
  - Disk
- Monitoring CPU utilization by thread to identify 'runaway' applications
- Perform actions on subsystems to correct/alleviate conditions
- Reconfigure systems to assist with load balancing

12.7.4.1 Monitoring Systems and Isolating Performance Problems

In monitoring performance as well as isolating potential problems, it is key to be able to receive notifications as thresholds on performance related resources are reached. It must also be possible to continually monitor or GET values of key performance indicators such as counters and timers.

The list below shows some of the information that may be monitored through the LAN NetView agents.

- The OS/2 CPU : Number of interrupts, CPU idle time
- The OS/2 Memory : Swapping statistics, number of page faults
- The OS/2 Disk : Cache utilization
- The OS/2 File I/O : Number of open files
- Disk I/O : Number of bytes read/written from disk.
- Logical serial port : Time spent reading/writing
- OS/2 thread : CPU used by thread, time spent waiting.
- Printer I/O : Number of write operations
- LAN Server : The activity of the server in read and write operations, use of buffers

As key performance indicators are tracked, management programs can be alerted as thresholds are approached.

A performance management application could then utilize operations and configuration management functions to perform corrective actions such as reconfiguration and/or load balancing.

12.7.4.2 Performance Management Summary

It was not feasible to list here all of the attributes that can be used in performance management. However, the MIB provides access to a large number of counters, timers, control blocks and other information that will allow for powerful performance management applications to be written.

12.7.5 Inventory Management

Tracking inventory in a large network can be a difficult task.

Workstations are constantly being added, removed, upgraded and reconfigured. Software inventory can be even more dynamic as different service levels of programs may exist and be applied independently.

Examples of Inventory Management

The following are examples of the types of inventory management tasks that may be desired in a distributed environment:

- Collecting software vital product data from known systems
- Collecting hardware vital product data from known systems
- Discovery of new nodes entering the network

An inventory management program can retrieve vital product data for all subsystems supported by the LAN NetView agents.

Information can be gathered for software and hardware. The list below provides examples of the type of information available through the LAN NetView agents that can be used for inventory management.

- Software
  - The product name, version and the CSD of the Operating System
  - The product name, version and the CSD of the OS/2 LAN Server, the computer name and the domain name.
  - The product name, version and the CSD of the Database Manager and of the Communication Manager
- Hardware
  - The identifier of the machine specified by the manufacturer
  - The Machine type, location, owner, contact information : this information is specified during system installation.
  - The display type
  - The CPU type, the co-processor type
  - The fixed disk and diskette drive size and capacity
  - The keyboard identifier and type
  - The microchannel adapter identifier

12.7.5.1 Discovery of New Nodes

The LAN NetView Manage product provides topology and discovery services. If new nodes that enter the network contain the LAN NetView agents, they will be identified to the managing system. Other nodes, such as those running TCP/IP will also be discovered by Manage. An inventory management application residing on the managing system could then query the software and hardware as described above to add the information to an inventory database.

12.7.5.2 Inventory Management Summary

The LAN NetView agents provide access of both hardware and software vital product data to management applications. This information is key to the development and maintenance of an inventory management database.

12.7.6 Summary of Agent Based Solutions

This chapter provided a glimpse of the capabilities of the LAN NetView MIB represented by the LAN NetView Agents and Agents Extended products.

However, to fully appreciate all of the building blocks that have been made available for systems management applications using the LAN NetView agents, we refer you to the LAN NetView product documentation.

12.8 LAN NetView MIB Summary

This section presents six tables that summarize the LAN NetView MIB.

This is NOT a complete reference of the LAN NetView MIB. Rather it is intended to provide the reader with an overview of the organization of the MIB and examples of the objects, attributes, actions and notifications it defines. Please refer to the LAN NetView product documentation for complete details on the MIB.

The tables are organized as follows:

Table 1 OS/2 Agent

Table 2 DOS Agent

Table 3 DOS/Windows Agent

Table 4 LAN Server Agent

Table 5 Communications Manager Agent

Table 6 Database Manager Agent

Each table lists the defined objects for the particular agent, a summary of the attributes, actions and notifications. Under the summary of attributes the following information will be presented:

- The number of attributes defined for the object
- The number of attributes the object has inherited from other classes (this is in addition to the number defined).
- Examples of attributes by group, if applicable.

Common Objects

It should be observed that the OS/2, DOS and DOS/Windows agents include many common object definitions. This allows management applications to easily access these different platforms and present the common information without requiring platform specific code.

12.8.1 OS/2 Agent

The OS/2 Agent is part of the LAN NetView Agents product. It defines 20 object classes that represent OS/2 objects as well as related hardware.

Table 12 (Page 1 of 7). Managed Objects for OS/2 AGENT

Object	Summary of attributes	Actions	Notifications
OS/2	Number of defined attributes : 68 Number of additional class defined attributes accessed by monitor-scanner agent : 59 Number of inherited attributes : 39 • Attribute groups – Session parameters Examples: maximum # of VDM sessions, maximum # of PM sessions – Memory information Example: page size, total available memory, total physical memory – Current system information Examples: boot drive, maximum path length, time date – Current country information Examples: code page, country code – Initialization parameters These attributes will return values of parameters in CONFIG.SYS such as Device, DOS, Buffers, DiskCache • Other attributes – Attributes for performance monitoring of memory, CPU, disk, file I/O. Examples: number of page faults, number of interrupts raised, number of files open. • Inherited attributes – Product attributes Examples: product name, product version, current CSD level – State/status Examples: usage state, operational state, availability status – Agent information Examples: agent version, start time, heartbeat rate	Restart system Shutdown system	Object creation Object deletion Attribute value change State change Agent Heartbeat

Table 12 (Page 2 of 7). Managed Objects for OS/2 AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Machine	Number of defined attributes : 15 Number of inherited attributes : 29 • Defined attributes – Information entered by the user during the installation Examples: the contact, the location, the owner, the type, the serial number – Information obtained from the machines ROM : the model ID • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Physical Keyboard	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Keyboard type and ID • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Display	Number of defined attributes : 4 Number of inherited attributes : 29 • Defined attributes Adapter memory, adapter type, display ID, display type • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Micro Channel* Adapter	Number of defined attributes: 1 Number of inherited attributes : 30 • Defined attribute Adapter ID • Inherited attributes State/status Examples: usage state, operational state, availability status Slot ID		Object creation Object deletion State change

Table 12 (Page 3 of 7). Managed Objects for OS/2 AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Floppy Drive	Number of defined attributes : 2 Number of inherited attributes : 30 • Defined attributes Floppy capacity and floppy size • Inherited attributes State/status Examples: usage state, operational state, availability status Physical storage ID		Object creation Object deletion State change
Fixed Disk Drive	Number of defined attributes : 5 Number of additional class defined attributes accessed by monitor-scanner agent : 10 Number of inherited attributes : 30 • Defined attributes Fixed disk capacity Number of sectors per cylinder, number of cylinders on a fixed disk • Inherited attributes State/status Examples: usage state, operational state, availability status Physical storage ID		Object creation Object deletion State change
Processor	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Processor Id, processor type • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Co-processor	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Co-processor Id, co-processor type • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change

Table 12 (Page 4 of 7). Managed Objects for OS/2 AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Logical Parallel Port	Number of defined attributes : 3 Number of additional class defined attributes accessed by monitor-scanner agent : 3 Number of inherited attributes : 31 • Defined attributes Infinite retry (on/off), port status (timeOUT,ioError), port timeOut value (seconds) • Inherited attributes State/status Examples: usage state, operational state, availability status Logical device driver, logical device name		Object creation Object deletion State change
Logical Serial Port	Number of defined attributes : 13 Number of additional class defined attributes accessed by monitor-scanner agent : 8 Number of inherited attributes : 31 • Defined attributes Communication status, modem input/output signal, Min/max BitRate, read/write TimeOut • Inherited attributes State/status Examples: usage state, operational state, availability status Logical device driver, logical device name		Object creation Object deletion State change
Logical Pointing Device	Number of defined attributes : 1 Number of inherited attributes : 31 • Defined attribute pointing device info • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Logical Volume	Number of defined attributes : 11 Number of inherited attributes : 31 • Defined attributes Local or remote, volume available space, volume name, volume size, volume serial number • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change

Table 12 (Page 5 of 7). Managed Objects for OS/2 AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Open File	Number of class defined attributes accessed by monitor-scanner agent : 7 Number of inherited attributes : 4 **Note: This object class is only accessible through the monitor-scanner object.** • Defined attributes File Name, time spent reading and writing to the file, Number of reads and writes to the file Number of bytes read from or written to the file • Inherited attributes State/status Examples: usage state, operational state, availability status		
OS/2 Process	Number of defined attributes : 8 Number of inherited attributes : 31 • Defined attributes Process Status, parent process Id, process type, number of thread control blocks Number of run-time libraries, list of run-time linked libraries • Inherited attributes State/status Examples: usage state, operational state, availability status, process ID, process name		Object creation Object deletion State change
Monitored File	Number of defined attributes : 7 Number of inherited attributes : 29 • Defined attributes Monitored file name, file size, time stamp, file Attributes, CRC (the signature of the file) • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change Processing error alarm

Table 12 (Page 6 of 7). Managed Objects for OS/2 AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
OS/2 Thread	Number of defined attributes : 6 Number of additional class defined attributes accessed by monitor-scanner agent : 6 Number of inherited attributes : 31 • Defined attributes Thread state, user time, system time, thread slot, sleep Id • Inherited attributes State/status Examples: usage state, operational state, availability status, thread ID, thread priority		Object creation Object deletion State change
Spooled Printer	Number of defined attributes : 10 Number of inherited attributes : 29 • Defined attributes Printer status, printer name, current print job Id, List of installed device drivers supported and list installed device drivers Time printing • Inherited attributes State/status Examples: usage state, operational state, availability status	Continue Delete current job Pause Restart job	Object creation Object deletion State change
OS/2 Spooler Queue	Number of defined attributes : 17 Number of inherited attributes : 29 • Defined attributes Connected printers, device driver, queue priority Separator file, queue start time, queue until time List queue processors, queue name, queue comment, Queue description, queue priority, queue status, queue type • Inherited attributes State/status Examples: usage state, operational state, availability status	Hold queue Release queue	Object creation Object deletion State change

Table 12 (Page 7 of 7). Managed Objects for OS/2 AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
OS/2 Spooler Job	Number of defined attributes : 15 Number of inherited attributes : 29 • Defined attributes Job priority Job Id, job size, document name, data type, job status Device driver data, device driver name • Inherited attributes State/status Examples: usage state, operational state, availability status	Copy the job Release the job Hold the job	Object creation Object deletion State change

12.8.2 DOS Agent

The DOS Agent is part of the LAN NetView Agents product. It defines 13 object classes that represent DOS objects and related hardware.

Table 13 (Page 1 of 4). Managed Objects for DOS AGENT

Object	Summary of attributes	Actions	Notifications
DOS	Number of defined attributes : 30 Number of inherited attributes : 39 • Attribute groups Current DOS country information Examples: code page, country code • Other attributes Memory management Examples: allocation strategy, extended memory, expended memory, total physical memory, total available memory Boot drive, net status, machine name, set version Printer setup, current tasks, link flag • Inherited attributes – Product attributes Examples: product name, product version, current CSD level – State/status Examples: usage state, operational state, availability status	Restart Shutdown	Object creation Object deletion State change Agent heartbeat
Machine	Number of defined attributes : 15 Number of inherited attributes : 29 • Defined attributes – Information entered by the user during the installation Examples: the contact, the location, the owner, the type, the serial number – Information obtained from the machines ROM : the model ID • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Physical Keyboard	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Keyboard type and ID • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change

Table 13 (Page 2 of 4). Managed Objects for DOS AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Display	Number of defined attributes : 4 Number of inherited attributes : 29 • Defined attributes Adapter memory, adapter type, display ID, display type • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Micro Channel* Adapter	Number of defined attributes: 1 Number of inherited attributes : 30 • Defined attribute Adapter ID • Inherited attributes State/status Examples: usage state, operational state, availability status Slot ID		Object creation Object deletion State change
Floppy Drive	Number of defined attributes : 2 Number of inherited attributes : 30 • Defined attributes Floppy capacity and floppy size • Inherited attributes State/status Examples: usage state, operational state, availability status Physical storage ID		Object creation Object deletion State change
Fixed Disk Drive	Number of defined attributes : 5 Number of additional class defined attributes accessed by monitor-scanner agent : 10 Number of inherited attributes : 30 • Defined attributes Fixed disk capacity Number of sectors per cylinder, number of cylinders on a fixed disk • Inherited attributes State/status Examples: usage state, operational state, availability status Physical storage ID		Object creation Object deletion State change

Table 13 (Page 3 of 4). Managed Objects for DOS AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Processor	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Processor Id, processor type • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Co-processor	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Co-processor Id, co-processor type • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Logical Parallel Port	Number of defined attributes : 3 Number of additional class defined attributes accessed by monitor-scanner agent : 3 Number of inherited attributes : 31 • Defined attributes Infinite retry (on/off), port status (timeOUT,ioError), port timeOut value (seconds) • Inherited attributes State/status Examples: usage state, operational state, availability status Logical device driver, logical device name		Object creation Object deletion State change
Logical Serial Port	Number of defined attributes : 13 Number of additional class defined attributes accessed by monitor-scanner agent : 8 Number of inherited attributes : 31 • Defined attributes Communication status, modem input/output signal, Min/max BitRate, read/write TimeOut • Inherited attributes State/status Examples: usage state, operational state, availability status Logical device driver, logical device name		Object creation Object deletion State change

Table 13 (Page 4 of 4). Managed Objects for DOS AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Logical Pointing Device	Number of defined attributes : 1 Number of inherited attributes : 31 • Defined attribute pointing device info • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change
Logical Volume	Number of defined attributes : 11 Number of inherited attributes : 31 • Defined attributes Local or remote, volume available space, volume name, volume size, volume serial number • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change

12.8.3 DOS/Windows Agent

The DOS/Windows Agent is part of the LAN NetView Agents product. It defines 3 object classes that represent Windows managed resources.

Table 14. Managed Objects for WINDOWS AGENT

Object	Summary of attributes	Actions	Notifications
Microsoft Windows	Number of defined attributes : 52 Number of inherited attributes : 39 • Attributes groups Startup country information Examples: init time format, init country code, init measurement system • Other parameters Total memory, initial virtual memory, Initial OEM font file, initial network driver, initial display driver • Inherited attributes – Product attributes Examples: product name, product version, current CSD level – State/status Examples: usage state, operational state, availability status	Shutdown	Object creation Object deletion Attribute value change State change Agent Heartbeat
Windows Spooler	Number of defined attributes : 4 Number of inherited attributes : 29 • Defined attributes Examples: default printer, printer drivers, print manager spooler name		Object creation Object deletion State change
Process	Number of defined attributes : 2 Number of inherited attributes : 29 • Defined attributes Process Id, process name • Inherited attributes State/status Examples: usage state, operational state, availability status		Object creation Object deletion State change

12.8.4 LAN Server Agent

The LAN Server Agent will be split between products. The object classes required to manage a LAN Requester will be provided as part of the LAN NetView Manage and Enabler products. The object classes required to manage a LAN Server will be provided as part of the LAN NetView Agents Extended product. There are 4 object classes representing the LAN Server, Requester and the connections between them.

Table 15 (Page 1 of 4). Managed Objects for LAN SERVER AGENT

Object	Summary of attributes	Actions	Notifications
LAN Server	Number of defined attributes : 61 Number of additional class defined attributes accessed by monitor-scanner agent : 50 Number of inherited attributes : 45 • Attribute groups – Server information Examples: component ID, current CSD level, product name, product version, run-time computer name, run-time domain name – Server initial system configuration Examples: network I/O ratio, The free space of the disk before before notifying the administrator The number of invalid I/O before notifying the administrator – Server run-time system configuration Examples: network I/O ratio, The free space of disk before notifying administrator The number of invalid logons before notifying administrator – Server run-time capacity configuration Examples: the maximum audit file size, the maximum files locks active number of connections to netnames allowed – Server run-time tunable config Examples: the configuration of buffers, heuristics – Server statistics Examples: the number of server file and named pipe opens, The number of server session starts, auto-disconnects, errored out, violations, access permission errors.	Clear statistics Pause Continue Activate Deactivate	Quality of service alarm Equipment alarm Processing error alarm Environmental alarm FFST Error Log Entry State change Object creation Object deletion

Table 15 (Page 2 of 4). Managed Objects for LAN SERVER AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
LAN Server	• Other attributes – Entry package attributes Examples: number of user sessions with the server, configuration of requester buffers, number of different reads / writes (small, multiplex, raws). – Advanced package attributes Examples: number of transactions processed by the server, configuration of requester buffers, number of different reads/writes (small, multiplex, raws). • Attributes inherited Product name, product version, current CSD level		

Table 15 (Page 3 of 4). Managed Objects for LAN SERVER AGENT

Object	**Summary of attributes**	**Actions**	**Notifications**
LAN Requester	Number of defined attributes : 63 Number of additional class defined attributes accessed by monitor-scanner agent : 25 Number of inherited attributes : 44 • Attribute groups – Requester information Examples: component ID, current CSD level, product name product version, run-time computer name, run-time domain name, user name – Requester run-time system configuration Examples: initial services to start, list names of networks on which the requester runs – Requester run-time capacity configuration Examples: the number of services that can be start on the requester, the number of buffers allocated for receiving datagrams. – Requester run-time tunable config Examples: the configuration of buffers, heuristics – Requester statistics Examples: the number of NCB issued (redirecter, server, application) The number of NCB failed (redirecter, server, application) – Other attributes The name of the primary domain controller for this machine If the redirecter is paused, if the redirecter for disks is paused If the redirection for spool device is paused – Attributes inherited Product name, product version, current CSD level	Clear statistics Pause Continue Activate Deactivate	Quality of service alarm Equipment alarm FFST Error Log entry Processing error alarm State Change Object creation Object deletion
Server Connection	Number of defined attributes : 8 Number of inherited attributes : 9 • Defined Attributes The name of the user at the client machine that made the session, The number of seconds a session has been active, idle The number of connections that have been made during the session • Attributes inherited Partner connection, connection ID		

Table 15 (Page 4 of 4). Managed Objects for LAN SERVER AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
IBMLAN.INI Configuration	Number of defined attributes : 3 Number of inherited attributes : 4 • Defined Attributes The name of the IBMLAN.INI file Time of last modification to IBMLAN.INI Time of last modification to backup IBMLAN.INI file.	Get/Set/Add/Delete parameter to IBMLAN.INI Enumerate/Add/Delete section to IBMLAN.INI Backup/Restore IBMLAN.INI	

12.8.5 Communications Manager Agent

The Communications Manager agent is part of the Agents Extended product. It defines 14 object classes for use by management applications.

Table 16 (Page 1 of 4). Managed Objects for COMMUNICATION MANAGER AGENT

Object	Summary of attributes	Actions	Notifications
Comm. Manager	Number of defined attributes : 4 Number of inherited attributes : 39 • Defined attributes Default configuration file, agent refresh rate, procedural status, SNA Node • Inherited attributes Product version, product name, current CSD level, prior CSD level Usage state, operational State, availability status Activate, deactivate	Activate Deactivate Deactivate when no users	Object creation Object deletion State change
Config. File	Number of defined attributes : 3 Number of inherited attributes : 4 • Defined attribute Configuration file name, is active, is default		
APPN* Network Node	Number of defined attributes : 2 Number of inherited attributes : 36 • Defined attributes Examples: Node capabilities • Inherited attributes Examples: usage state, operational state, availability status SNA node name, dependent LU list	Activate Deactivate Deactivate when no users Activate with parameters Deactivate with parameters	State change Communications alarm Object creation Object deletion FFST Error Log entry
APPN End Node	Number of defined attributes : 1 Number of inherited attributes : 36 • Defined Attributes Network node server pointer • Inherited attributes Examples: usage state, operational state, availability status SNA node name, dependent LU list	Activate Deactivate Activate with parameters Deactivate with parameters Deactivate when no users	State change Communications alarm Object create Object delete FFST Error Log entry

Table 16 (Page 2 of 4). Managed Objects for COMMUNICATION MANAGER AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Logical Unit 6.2	Number of defined attributes : 3 Number of inherited attributes : 35 • Defined Attributes Default LU, active partner LU list, LU alias • Inherited attributes Usage state, operational State, availability status The destination network address of the LU for session, Name of the PU that's supporting this dependent LU.	Activate Deactivate Deactivate with parameters Deactivate when no users	Object creation Object deletion State change
Partner LU	Number of defined attributes : 8 Number of inherited attributes : 4 • Defined Attributes Examples: Partner LU alias, partner LU name, active mode list • Inherited attributes Object class , package		
Active Mode	Number of defined attributes : 17 Number of inherited attributes : 4 • Defined Attributes Examples: Mode name, name of the class of service, The minimum number of contention loser/winner session permissible between the local LU and the partner LU. maximum RU size upper/lower		
Transaction Program	Number of defined attributes : 4 Number of inherited attributes : 30 • Defined Attributes Examples: transaction program ID, name The conversations in which the TP is involved Local or remote initiated Queue depth • Inherited attributes Object class , package, operational state, usage state	Change back Change over Activate Deactivate	Attribute value Change Object creation Object deletion State change

Table 16 (Page 3 of 4). Managed Objects for COMMUNICATION MANAGER AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
LU 6.2 Session	Number of defined attributes : 2 Number of inherited attributes : 51 • Defined Attributes Examples: session ID, session type Type of pacing, send/receive RU size, send/receive pacing size destination address field, origin destination address indicator origin address field • Inherited attributes Object class , package, operational state, usage state Partner connection, connection ID	Activate Deactivate Deactivate with parameters	Object creation Object deletion State change
Conversation	Number of defined attributes : 14 Number of inherited attributes : 35 • Defined Attributes Examples: conversation state, conversation type, Local identifier of the resource Conversation group ID, the synchronization level of the conversation Name of the source/target TP Logical Unit of Work ID Conversation security • Inherited attributes Object class , package, operational state, usage state Partner connection, connection ID	Activate Deactivate Deactivate with parameters	Object creation Object deletion State change
APPN Transmission Group	Number of defined attributes : 10 Number of inherited attributes : 39 • Defined Attributes Examples: CP-CP session support, propagation delay Cost per connect time, cost per byte transmitted, User defined weighting, the security of the TG. • Inherited attributes Object class , package, operational state, usage state Partner connection, connection ID	Activate Deactivate	Object creation Object deletion State change

Table 16 (Page 4 of 4). Managed Objects for COMMUNICATION MANAGER AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
Logical Link	Number of defined attributes : 21 Number of inherited attributes : 35 • Defined Attributes Examples: link name, adapter numbers, port ID, adjacent node type and name, active sessions Line type, adjacent link station name and address • Inherited attributes Object class , package, operational state, usage state Partner connection, connection ID	Activate Deactivate Deactivate when no user Deactivate with parameters	Object creation Object deletion State change
Port	Number of defined attributes : 16 Number of inherited attributes : 35 • Defined Attributes Examples: the name of the local DLC, port ID, adapter numbers, adapter address send/receive window size, line type, link station role • Inherited attributes Object class , package, operational state, usage state Partner connection, connection ID	Activate Deactivate Deactivate with parameters Deactivate when no user	Object creation Object deletion State change
Peer Tree	Number of defined attributes : 2 Number of inherited attributes : 4 • Defined Attributes Peer tree instance , Peer tree name		Object creation Object deletion

12.8.6 Database Manager Agent

The Database Manager agent is part of the Agents Extended product. It defines 6 object classes for use by management applications.

<table>
<caption>Table 17 (Page 1 of 2). Managed Objects for DATABASE MANAGER AGENT</caption>
<tr><th>Object</th><th>Summary of attributes</th><th>Actions</th><th>Notifications</th></tr>
<tr><td>Database</td><td>Number of defined attributes : 38
Number of inherited attributes : 38
Attribute groups
Database configuration view
Examples: code page, if the database is in consistent state
If the roll-forward recovery is in progress
Database configuration update
Examples: the size of the logs, the path of the log file
Percentage of lock list allowed per application
Database status
Examples: the time of the last backup, the number of users currently connected to the database, name of the database resource
Inherited attributes
Examples: Object class , package, operational state, usage state
Resource alias, resource name</td><td>Activate
Deactivate
Backup the DB
Reset the DB configuration
Restart the database
Restore the database
Roll Forward the database</td><td>Object creation
Object deletion
State change</td></tr>
<tr><td>Database Manager</td><td>Number of defined attributes : 10
Number of inherited attributes : 36
Attribute groups
DBM configuration view
Examples: node type,product version, product name
DBM configuration update
Examples: the name of the workstation, number of concurrent active database allowed
Inherited attributes
Examples: Object class , package, operational state, usage state
Product name, product version, current CSD level</td><td>Reset DBM configuration
Activate
Deactivate</td><td>Object creation
Object deletion
State change
Processing error alarm
FFST Error Log entry</td></tr>
</table>

Table 17 (Page 2 of 2). Managed Objects for DATABASE MANAGER AGENT			
Object	**Summary of attributes**	**Actions**	**Notifications**
DB Directory Entry	Number of defined attributes : 6 Number of inherited attributes : 8 • Defined attributes Examples: name of the drive and name of the node where the database resides If the database is remote or local • Inherited attributes Examples: Object class , package, DBMS product type Resource alias, resource name		Object creation Object deletion
Remote GW Directory Entry	Number of defined attributes : 5 Number of inherited attributes : 8 • Defined attributes Examples: DLL name of the application client, transaction program prefix SQLCODE mapping file name, resource name, resource alias • Inherited attributes Examples: Object class , package, DBMS product type Resource alias, resource name		Object creation Object deletion
Remote Node Directory Entry	Number of defined attributes : 10 Number of inherited attributes : 4 • Defined attributes Examples: remote node name, protocol, adapter, mode, partner LU • Inherited attributes Examples: Object class , package		Object creation Object deletion
DB Gateway	Number of defined attributes : 1 Number of inherited attributes : 36 • Defined attributes A role attribute of underlying DBMS • Inherited attributes Product name, product version, current CSD level, prior CSD level Usage status, operational state		Object creation Object deletion State change

Chapter 13. Using a LAN NetView Managed-Object Catalog

This paper provides a general overview and description of an IBM LAN NetView managed-object catalog.

This paper is intended for IBM marketing representatives, IBM system engineers, customers, and software developers who desire a general understanding of the managed-object catalogs provided by the IBM LAN NetView family of products. Its purpose is to help the reader to become familiar with what an IBM LAN NetView managed-object catalog is and what it contains. It takes the reader through an IBM LAN NetView managed-object catalog, pointing out how to determine what objects, attributes, actions, etc. are available to a user or an application developer. It also relates the information in a managed-object catalog to the managed-object definition templates, the management content package definitions and encoding definition description files, and the C header files, which are shipped with the IBM LAN NetView Manage 1.0 development toolkit.

13.1 Introduction

This paper provides a general overview and description of an IBM LAN NetView managed-object catalog and intends to help the reader to become familiar with what an IBM LAN NetView managed-object catalog is and what it contains. It takes the reader through a managed-object catalog, pointing out how to determine what objects, attributes, actions, etc. are available to users or software developers. It uses a set of simple but typical scenarios to illustrate where the information is and how to find it. It also relates the information in a managed-object catalog to the managed-object templates, the management content package definitions and encoding definition description files, and the C header files, which are shipped with the IBM LAN NetView Manage 1.0 development toolkit.

After reading this paper, the reader will become familiar with what a managed-object catalog contains and how to use it with the IBM LAN NetView products.

13.2 Reader's Background Knowledge

A managed-object catalog contains information for users of the IBM LAN NetView family of products and management application developers using the IBM LAN NetView Manage product. The reader is assumed to have the following background in order to fully understand the information provided.

- General networking familiarity. In particular, readers should be conversant with the following concepts:
  - Internet and Open Systems Interconnection (OSI) addressing schemes
  - Concepts and practical aspects of modern computer networks, including network devices such as bridges and routers
  - Network topologies, such as bus, star, and rings
- Substantial knowledge of network management principles, the Common Management Information Protocol (CMIP) for OSI networks, the Simple

Network Management Protocol (SNMP) for TCP/IP networks, and related topics. The readers are also assumed to have some familiarity with the OSI network management framework and have access to the standard books.

- A general understanding of object-oriented principles and design techniques. Although experience with object-oriented programming is not required, an understanding of object-oriented concepts is necessary.
- A general understanding of the templates and other notational conventions defined in *ISO 10165-4, Management Information Services -- Structure of Management Information Part 4: Guidelines for the Definition of Managed Objects.*
- A general understanding of XOM/XMP API as described in the *IBM LAN NetView Manage and Enabler Developer's Guide (S96F-8493)* or *IBM LAN NetView Manage and Enabler Developer's Reference (S96F-8494).*

13.3 About an IBM LAN NetView Managed-Object Catalog

An IBM LAN NetView Managed-Object Catalog is a repository of GDMO templates for a specific category of managed-object definitions that can be used with the IBM LAN NetView Agents Extended, the IBM LAN NetView Agents for DOS, the IBM LAN NetView Manage, and the IBM LAN NetView Enabler products to manage the resources of the following software products:

- IBM Operating System/2
- IBM DOS 5.0/6.1
- Microsoft DOS 5.0/6.0
- Microsoft Windows 3.0/3.1
- IBM LAN Server 3.0
- IBM LAN Requester 3.0
- IBM Communication Manager/2 2.0
- IBM Extended Services 1.0 Database Manager
- IBM Database 2 for OS/2 (DB2/2) 1.0

The IBM LAN NetView product library includes the following managed-object catalogs:

- *IBM LAN NetView Operating Systems Agents Managed-Object Catalog*, (S96F-8495)
- *IBM LAN NetView Agents Extended Communication Manager Managed-Object Catalog*, (S96F-8497)
- *IBM LAN NetView Agents Extended Database Manager Managed-Object Catalog*, (S96F-8496)
- *IBM LAN NetView Agents Extended LAN Server Managed-Object Catalog*, (S96F-8498)

Other IBM publications that can be helpful along with these catalogs include:

- *IBM LAN NetView Manage Administration Guide*, (S96F-8492)
- *IBM LAN NetView Manage and Enabler Developer's Reference*, (S96F-8494)
- *IBM LAN NetView Manage and Enabler Developer's Guide*, (S96F-8493)

13.4 General Organization of a Managed-Object Catalog

The content of each Managed Object Catalog varies, depending on the types of managed-object definitions it describes. In general, a managed-object catalog is organized as follows:

1. The first chapter, *Introduction*, provides general information about the managed-object definition in the managed-object catalog:

   - an enumeration of the managed-object classes in the catalog
   - inheritance and naming hierarchies
   - naming examples for the managed-object classes
   - description of notations used in the managed-object class tables

2. The next series of chapters (that is, Chapters 2-29 of the *IBM LAN NetView Operating systems Agents Managed-Object Catalog* , S96F-8495-00) provide information about the managed-object classes defined for the appropriate IBM LAN NetView Agents Extended or IBM LAN NetView Agents for DOS agent. These chapters are arranged in alphabetical order by class name. Each of these chapters provides information about a specific managed-object class defined for a specific IBM LAN NetView agent:

   - A table that lists the characteristics inherited by this managed-object class. These inherited characteristics include attributes, attribute groups, notifications, actions, etc. The table provides the name of the managed-object class which introduces the inherited characteristics and the name of the document where the inherited characteristics are defined.

   - Another table that provides the names, descriptions, and access about the characteristics defined by the managed-object class. These class-defined characteristics include attributes, attribute groups, notifications, actions, etc.

   - Information specific to the current implementation of the managed-object class.

3. The last chapter, *GDMO Templates and ASN.1 Modules*, provides the GDMO templates and ASN.1 modules for the managed-object definitions in the catalog. In this chapter, the GDMO templates are grouped by types (see below). Within each group, the templates are arranged in an alphabetical order by template label.

   The following types of GDMO templates are included:

   - **Managed-object class** templates

     Each of these templates forms the basis of the formal definition of a managed object. Elements in a template allow the class to be placed at the appropriate node of the inheritance tree, and specify the various characteristics and the behavior of the class. This template contains major elements including:

     - complete specification of the behavior of the class
     - name of the superclass from which this class has inherited
     - mandatory and/or conditional packages
     - class naming
     - object identifier for the class template

   - **Package** templates

Each of these templates allows a package consisting of a combination of behavior definitions, attributes, attribute groups, operations, notifications, and parameters to be defined for subsequent inclusion into a **Managed-object class** template. This template contains major elements including:

- complete specification of the behavior of this package
- a set of attributes and/or attribute groups contained in the package
- specification of attribute-related operations
- specification of notifications that this package shall be able to generate
- object identifier for the package template

• **Parameter** templates

Each of these templates permits specification and registration of parameter syntaxes and associated behavior that may be associated with particular attributes, operations, notifications within the **Package**, **Attribute**, and **Notification** templates. This template contains major elements including:

- complete specification of the behavior of this parameter
- definition of processing failures
- specification of parameters of action requests/responses
- specification of parameters of notification requests/responses
- object identifier for the parameter template

• **Name Binding** templates

Each of these templates allows an attribute to be selected as the naming attribute that shall be used when a subordinate object, which is an instance of a specified managed-object class, is named by a superior object, which is an instance of a specified managed-object class or other object class. This allows an instance of a specified managed-object class to be placed at the appropriate node of a naming tree. The object identifier and the value assigned to the naming attribute are used to build the relative distinguished name (RDN) for the managed-object class where the naming attribute is defined.

• **Attribute** templates

Each of these templates is used to define individual attributes types. These definitions may be further combined by the **Attribute Group** template where attribute groups are required. This template contains major elements including:

- complete specification of the behavior of this attribute
- syntax that may convey the value of the attribute
- value matching, that is, whether the attribute may be tested for equality, magnitude, etc.
- specification of attribute-specific error parameters associated with management operations on the attribute
- object identifier for the attribute template

• **Attribute Group** templates

Each of these templates allows attribute groupings to be defined; such groupings are applicable to situations where it is desirable to operate upon the collection of attributes that are members of the group. This template defines the minimum set of attributes that constitute the group and the object identifier value that is used to name the group.

- **Behavior** templates

  Each of these templates is used to define behavioral aspects of managed-object class, name bindings, parameters, attributes, action, and notification types.

- **Action** templates

  Each of these templates is used to define the behavior and syntax associated with a particular action type. This template contains major elements including:

  - complete specification of the behavior of this action
  - specification of the mode of operation, that is, whether this action is *confirmed* or *unconfirmed*
  - abstract syntax that may convey the action information and action reply parameters
  - object identifier for the action template

- **Notification** templates

  Each of these templates is used to define the behavior and syntax associated with a particular Notification type. This template contains major elements including:

  - complete specification of the behavior of this notification
  - definition of event information, reply parameters, or error parameters associated with this notification
  - abstract syntax that may convey the event information and event reply parameters
  - object identifier for the notification template

Refer to *ISO/IEC 10165-4 Information Technology - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects* for the detailed specification of structure, syntactic definition, and semantic of each of the above types of templates.

Each catalog contains one or more ASN.1 modules for the managed objects defined in it. In the OSI standards, Abstract Syntax Notation One (ASN.1) is a formal description language used to define a set of primitive data types. This standard language defines a number of simple data types, with their tags, and specifies a notation for referencing these data types and for specifying values of these data types. In addition, ASN.1 provides a mechanism to construct new data types from the primitive data types already defined. A collection of ASN.1 descriptions to define the data types used by a set of managed objects based on a common theme is named an ASN.1 module.

Refer to *ISO/IEC 8824 Information Technology - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)* for the specification of the ASN.1 notation and data type definition.

4. A number of appendices provide some or all of the following:

   - GDMO templates and ASN.1 modules for the IBM LAN NetView Common Definitions
   - Summary of object identifiers of the object classes, attributes, attribute groups, actions, parameters, name binding, packages, relationship bindings, and relationship classes for the managed objects defined in a catalog
   - Summary of syntax types of the attribute values, notification information, action information, and parameter values defined in the catalog

- Summary of OM attributes
- Summary of value lists
- Summary of encoding definitions for the OM classes and attributes defined in the catalog
- Generic alerts mapped to ISO alarms
- Other information pertaining to a specific IBM LAN NetView agent

For detailed description of XOM and OM content package, refer to *IBM LAN NetView Manage and Enabler Developer's Guide*, S96F-8493 or *IBM LAN NetView Manage and Enabler Developer's Reference*, S96F-8494

13.5 Examples of Using a Managed-Object Catalog

The following sections use a set of scenarios to take the readers through an IBM LAN NetView managed-object catalog to point out how to read it to find the information pertaining to a scenario. In order to keep the description simple and easy to follow, the following scenarios are relevant to a specific IBM LAN NetView managed-object catalog. The same procedures may be applied to other catalogs.

Unless otherwise stated, all the references (that is, chapter numbers, table numbers, etc.) made hereinafter in this section are related to the *IBM LAN NetView Operating Systems Agents Managed-Object Catalog*,S96F-8495.

13.5.1 Managed-Object Classes

Scenario: To determine the managed-object classes supported by the IBM LAN NetView Operating System/2 agent.

Steps:

1. To find out all the managed-object classes supported by the OS/2 agent, refer to Table 1-1 on page 1-2 for a list of managed-object classes which can be instantiated by the OS/2 agent. The names of the instantiable classes are listed as follows:

   - *coprocessor*
   - *display*
   - *fixedDiskDrive*
   - *floppyDrive*
   - *logicalParralelPort*
   - *logicalPointingDevice*
   - *logicalSerialPort*
   - *logicalVolume*
   - *machine*
   - *microChannelAdapter*
   - *monitoredFile*
   - *openFile*
   - *operatingSystem2*
   - *os2Process*
   - *os2SpoolerJob*
   - *os2SpoolerQueue*
   - *os2Thread*
   - *physicalKeyboard*
   - *processor*
   - *spooledPrinter*

An *instantiable* class is a managed-object class of which an *instance* can be created by a process called *instantiation*. An instance is something you can do things to and has state, behavior, and identity. Instantiation is a process of filling in the templates of a managed-object class from which an instance of it can be created. An *abstract* class is a managed-object class that is never instantiated. Therefore, an abstract class has no instances and is defined such that its subclasses will add to its structure and behavior.

13.5.2 Naming Hierarchy for the Managed-Object Classes

Scenario: To determine the naming hierarchy defined for the managed-object classes which can be instantiated by the OS/2 agent.

Steps:

1. Refer to Figure 1-2 on page 1-5 for the naming hierarchy. Note that only the instantiable classes are included in the naming tree. Each instantiable class has a distinguished name (DN) and a relative distinguished name (RDN). For information about distinguished name, relative distinguished name, and the containment (naming) relationship, see the *IBM LAN NetView Manage and Enabler Developer's Guide*, S96F-8493.

2. To find out the individual class RDN for a managed-object class, see the **Name Binding** template for the class to determine the distinguishing attribute. For example, to determine the distinguishing attribute of the *machine* managed-object class, refer to the *system-machine* **Naming Binding** template on page 30-26. The distinguishing attribute is serial number, *serialNumber*. Now refer to the **Attribute** template for *serialNumber* on page 30-59. Note that the object identifier for the attribute *serialNumber* is {1 3 18 0 0 3315 1 7 102}. For the description of *serialNumber*, refer to the *serialNumberBehavior* **Behavior** template on page 30-132. If the serial number of an instance of the *machine* managed-object class is 123, then the distinguished name of the specific instance of this managed-object class is

   {0.0.13.3100.0.7.30=ORSM1, 2.9.3.2.7.4=D933E01,
   1.3.18.0.0.3315.1.7.102=123}

   where the RDN for the *networkId* attribute for the network named ORSM1 is {0.0.13.3100.0.7.30=ORSM1}, and the RDN for the system named D933E01 is {2.9.3.2.7.4=D933E01}.

13.5.3 Inheritance Hierarchy for the Managed-Object Classes

Scenarios: To determine the inheritance hierarchy for the managed-object classes defined for the IBM LAN NetView Operating System/2 agent.

Steps:

1. Refer to Figure 1-1 on page 1-4 for the inheritance hierarchy. Note that both the instantiable classes and the abstract classes are included in the inheritance tree. There are 11 abstract classes from which the other managed-object classes have inherited their characteristics.

   The following five abstract classes are defined in the main chapters of the *IBM LAN NetView Operating System Agents Managed-Object Catalog*, S96F-8495.

   - *adapter*
   - *logicalDevice*
   - *physicalStorage*

- *process*
- *thread*

The following two abstract classes are defined in the Appendix A of the *IBM LAN NetView Operating System Agents Managed-Object Catalog*, S96F-8495.

- *GCD: agent*
- *GCD: subsystemProduct*

The following four abstract classes are defined in the specified standard documents specified in Figure 1-1.

- *DMI: top*
- *GMI: subsystem*
- *GMO: function2*
- *OP1V4: function*

13.5.4 A Managed-Object Class

Scenario: To determine the characteristics defined in the *spooledPrinter* managed-object class.

Steps:

1. Refer to the table of contents of the managed-object catalog to determine the chapter where *spooledPrinter* managed-object class is defined. Note that the chapters are organized in an alphabetical order by class name. The *spooledPrinter* managed-object class is defined in Chapter 27 on page 27-1.
2. Refer to Chapter 27 which describes the characteristics and the current implementation of the *spooledPrinter* managed-object class.

   Table 27-1 lists the inherited characteristics by the *spooledPrinter* managed-object class. These characteristics are inherited from the following abstract managed objects:

   - *top*
   - *function*
   - *function2*

   Table 27-2 describes the characteristics which are defined for the *spooledPrinter* managed-object class. There are 10 attributes (for example, portName) and 4 actions (for example, pause).
3. Refer to Chapter 30 for the GDMO templates defined for the *spooledPrinter* managed-object class. For the **Managed-object class** template, refer to the one under the name *spooledPrinter* on page 30-4, which points to the *spooledPrinterPackage* **Package** template.
4. Refer to the *spooledPrinterPackage* **Package** template on page 30-21 about the attributes and actions defined for this managed-object class.
5. Refer to the *spooledPrinterBehavior* **Behavior** template on page 30-134 for the behavior of the *spooledPrinter* managed-object class.
6. Refer to Chapter 30 for the **Attribute** templates for the attributes defined.
7. Refer to Chapter 30 for the **Action** templates for the actions defined.

13.5.5 Single Attribute

Scenario: To find out more about the definition of a single attribute (for example, *portName* attribute of the *spooledPrinter* managed-object class).

Steps:

1. Refer to the *spooledPrinterPackage* **Package** template of the *spooledPrinter* managed-object class on page 30-21. One of the defined attributes is *portName*. The *portName* attribute is the only attribute of this managed-object class that supports both the GET and REPLACE operations.

   If a failure occurs within the agent while a request for either operation is being processed, an instance of the *spooledPrinter* managed-object class returns a processing failure error that contains information about the failure. This is accomplished by associating the *"LAN NetView Common Definitions : 1993": processingFailureInfo* parameter with this attribute. See Appendix A, *"LAN NetView Common Definitions : 1993"*, for these definitions. Refer to page A-5 for the *processingFailureInfo* **Parameter** template, which points to the *processingFailureInfoBhv* **Behavior** template on page A-10. The ASN.1 for *processingFailureInfo* is defined on page A-15.
2. Refer to the **Attribute** template for *portName* on page 30-53.
3. Refer to the **Behavior** template for *portName* on page 30-123.
4. About the ASN.1 data type defined for the attribute *portName*, refer to the OS-Attribute-ASN1Module on page 30-153. The ASN.1 data type for *portName* is GRAPHIC STRING.

13.5.6 Action

Scenario: To find out more about the definition of a single action (for example, action *deleteCurrentJob* of the *spooledPrinter* managed-object class).

Steps:

1. Refer to the *spooledPrinterPackage* **Package** template of the *spooledPrinter* managed-object class on page 30-21. One of the actions defined is *deleteCurrentJob*.
2. Refer to the **Action** template for *deleteCurrentJob* on page 30-142.
3. The behavior of this action is described by the *deleteCurrentJobBehavior* **Behavior** template on page 30-79.
4. If a failure occurs within the agent while a request for this action is being processed, an instance of the *spooledPrinter* managed-object class returns a processing failure error that contains information about the failure. This is accomplished by associating the *"LAN NetView Common Definitions : 1993": processingFailureInfo* parameter with this action. See Appendix A, *"LAN NetView Common Definitions: 1993"*, for these definitions.

13.5.7 Notification

Scenario: To find out about a notification defined by a managed-object class (for example, heartbeat notification generated by the *operatingSystem2* managed-object class).

Steps:

1. Refer to the definition of the *operatingSystem2* managed-object class in Chapter 18. This class has inherited five different notifications (see page 18-2). One of the inherited notifications is *heartbeatNotification* from *agent* managed-object class.
2. The *agent* managed-object class is defined in the LAN NetView Common Definitions: 1993 in Appendix A. Refer to page A-12 for the **Notification** template for the *heartbeatNotification*,
3. Refer to the *agentHeartBeatBehavior* **Behavior** template on page A-7, which describes three different types of heartbeat notification that can be generated by an IBM LAN NetView agent: initial, periodic, and terminal.
4. Refer to *os2AttributeModule* on page A-14 for the ASN.1 data type defined for *HeartbeatInfo*. It is of ENUMERATED type with possible values of 0 (initial), 1 (terminal), or 2 (periodic).

13.6 Relationship with the IBM LAN NetView Development Toolkit

The IBM LAN NetView development toolkit, which is included in the IBM LAN NetView 1.0 products, provides a set of .GDM, .TXT, and .H files for development of management application. After a toolkit is installed, the following files are available in these directories:

- drive:\LNV\ETC\MIB*.gdm
- drive:\LNV\ETC\TXT*.txt
- drive:\LNV\INCLUDE*.h

These files are related to the IBM LAN NetView managed-object catalogs.

13.6.1 GDMO Templates Files

The .GDM files in the drive:\LNV\ETC\MIB directory are the GDMO templates defined and supported by the IBM LAN NetView product family.

The GDMO templates for the managed-object classes defined and supported in the IBM LAN NetView Operating System/2 agents are provided in the following files:

- osa.gdm
- gcd.gdm
- gmi.gdm
- gmo.gdm
- dmi.gdm
- op1v4.gdm

These files are in ASCII format and contain the GDMO templates which are defined and referenced in the *IBM LAN NetView Operating Systems Agents Managed-Object Catalog*, S96F-8495.

13.6.2 OM Package Definitions and Encoding Definitions Description Files

The IBM LAN NetView development toolkit provides a set of files which describe the OM Content Package Definitions and Encoding Definitions, which are briefly described below. For detailed description of XOM and XMP, refer to Chapter 4 of *IBM LAN NetView Manage and Enabler Developer's Guide*, S96F-8493.

1. OM Content Package Definitions

The OM Content Package Definitions are divided into following five areas:

a. OM Package Object Identifier - provides the OSI object identifier assigned to the entire OM package that represents the information model (for example, IBM LAN NetView Operating Systems Agents managed-object catalog).

b. Object Identifier Tables - list the OSI object identifiers assigned to OSI managed-object classes, attributes (and attribute groups), notifications, actions, and parameters. Other object identifiers associated with the information model, such as name bindings or conditional packages, are not included because they do not affect the XOM/XMP API directly.

c. Information Syntax Tables - provide a list of OM classes which represent the syntax of an Attribute, Notification, Action, or Parameter. These tables resolve the ANY syntaxes which appear in the XMP API.

d. OM Attribute Tables - provide a list of OM attributes which represent the content of each OM class. For each OM attribute, the attribute's name, syntax, length and number of values are provided.

e. Value Lists - enumerate the values which can be assigned to OM attributes of the following types:

   - ENUM
   - INTEGER (when representing an ASN.1 named integer list)
   - STRING (Object-Identifier)
   - STRING (Bit)

2. Encoding Definitions

   The Encoding Definitions describe the type of encoding information required for the XMP API to perform the OM/BER encoding/decoding of the information model for the application. The Encoding Definitions can be divided into two areas: the OM Class Encoding Definitions and the OM Attribute Encoding Definitions.

The description of the OM Content Package Definitions and Encoding Definitions for the managed-object classes defined and supported in the *IBM LAN NetView Operating Systems Agents Managed-Object Catalog*, S96F-8495 are provided in the following files:

- osa.txt
- gcd.txt
- gmi.txt
- gmo.txt
- dmi.txt
- op1v4.txt

13.6.3 C Header Files

The C header files contains generic C-language **#define** statements representing the information model which can be used by an application using XMP API. Each of these files consists of the **#define** statements for the following:

- OM Package Object Identifiers
- Managed-Object Identifiers
- OM Class Object Identifiers
- Name Constants
- Value List

The C header files for the managed-object classes defined and supported in the *IBM LAN NetView Operating Systems Agents Managed-Object Catalog*, S96F-8495 are provided in the following files:

- osa.h
- gcd.h
- gmi.h
- gmo.h
- dmi.h
- op1v4.h

13.7 Summary

This paper provides a general technical overview of an IBM LAN NetView managed-object catalog.

In this simple exposition of facts, the reader should have acquired a general knowledge of what a managed-object catalog is and what it contains. A list of IBM LAN NetView managed-object catalog publications is provided for reference. The general organization of an IBM LAN NetView managed-object catalog is described so that the reader will have become familiar with how a catalog is structured before using it. This paper also briefly describes various OSI standard managed-object templates which are used in the IBM LAN NetView managed-object catalogs to define the characteristics of the managed-object classes. Then, this paper uses a set of simple scenarios to walk the reader through the *IBM LAN NetView Operating Systems Agents Managed-Object Catalog* to point out how to determine what information is contained in the catalog and how to find it.

Finally, the paper relates the information contained in the catalog with some of the files (that is, the GDMO template files, the management content package definitions and encoding definitions description files, and the C header files) which are shipped with the IBM LAN NetView 1.0 products.

Having read the paper, the reader should know how to read and use an IBM LAN NetView managed-object catalog.

Chapter 14. Accessing SNMP Managed Objects Using CMIP Services from the LAN NetView XMP API

This paper demonstrates how an application designer can use the CMIP services of LAN NetView's XMP API to access SNMP managed objects.

A sample program, which retrieves the contents of the **TcpConnTable** aggregate table defined in MIB-2, is included in 14.4, "Sample Code to Access tcpConnTable" on page 211.

14.1 Scope

This paper is aimed primarily at application developers who plan to implement systems/network management solutions on the IBM LAN NetView platform. It explains how one can use the CMIP services of LAN NetView's XMP API to access SNMP managed objects.

This paper assumes that the reader has a working knowledge of the X/Open OSI-Abstract-Data Manipulation (XOM) and X/Open Management Protocol (XMP) application programming interfaces, as used by the LAN NetView family of products. The reader should also be familiar with the object modeling paradigm offered by LAN NetView.

The following LAN NetView publications provide additional information:

- *LAN NetView Manage 1.0 Administration Guide*, S96F-8492
- *LAN NetView Manage and Enabler Version 1.0 Developer's Guide*, S96F-8493
- *LAN NetView Manage and Enabler Version 1.0 Developer's Reference*, S96F-8494

14.2 Accessing Managed Objects Using XMP

The LAN NetView XMP API provides two distinct services which may be used to access managed objects.

- SNMP services
- CMIP services

The SNMP services are used to access SNMP managed objects, while the CMIP services are used to access CMIP managed objects. Both services are tailored for a particular type of managed object. As one would expect, the CMIP services are more complicated and hence require more OM (OSI-Abstract-Data Manipulation) classes than do the SNMP services.

This section lists the function calls and the OM classes used to access a managed object (that is, perform a query).

14.2.1 SNMP Services

SNMP service are used to access SNMP managed objects only. These services provide the following access function calls:

- mp_get_req
- mp_get_next_req
- mp_get_rsp

The OM classes used for these function calls include the following:

- SNMP-Get-Argument
- SNMP-Get-Result
- SNMP-Response
- Var-Bind

14.2.2 CMIP Services

The CMIP services provide the following access function calls:

- mp_get_req
- mp_get_rsp

The OM classes used for these function calls include the following:

- CMIS-Get-Argument
- CMIS-Get-Result
- Scope
- CMIS-Filter
- Attribute-Id-List
- Attribute-Id
- Attribute
- Object-Class
- Object-Instance
- DS-DN
- DS-RDN
- AVA

14.2.3 Why Use the CMIP Service to Access SNMP Managed Objects

Using the SNMP services of the XMP API to access SNMP managed objects will be the most prevalent method chosen by application designers, because of its simplicity. However, application designers may wish to consider the CMIP services of the XMP API in accessing SNMP managed objects.

Using the CMIP services has some advantages as well as some drawbacks.

Advantages:

- If already familiar with CMIP services, the application designer\developer does not need to worry about learning the SNMP services.
- Ability to retrieve entire contents of aggregate tables (for example, tcpConnTable) with one single **mp_get_req** function call. To obtain the similar results with SNMP services, the application must invoke **mp_get_next** repeatedly until the results indicate that the last entry of the aggregate table has been retrieved.
- Allows application program to be portable since the application does not make use of the **mp_get_next_req** function call which is not valid for CMIP services.

Disadvantages:

- SNMP MIBs to be accessed must be translated[1] into CMIP and integrated using the **SNMPMIB** command.
- A local registration file (LRF) must be created for each SNMP MIB to be accessed.
- Since we are using the CMIP modeling paradigm, attributes to be retrieved must all be defined in the same managed object class.

  For example, one cannot retrieve both **sysDescr** and **tcpMaxConn** in one single **mp_get_req** invocation, since the two attributes are not defined in the same managed object class. The **sysDescr** attribute is defined under the **system** managed object class while the **tcpMaxConn** attribute is defined under the **tcp** managed object class.

14.2.4 How to Access SNMP Managed Objects Using CMIP Services

In order to access SNMP managed objects using CMIP services, the following process must be followed:

1. Determine if the SNMP managed object is defined in MIB-2 or whether it is an extension to MIB-2.
2. If the SNMP managed object is defined as a MIB-2 extension:
   - Obtain the file which defines the MIB-2 extension
   - Translate[2] the MIB file from SNMP nomenclature to CMIP
   - Integrate the translated MIB file by using the **SNMPMIB** command
3. Create a local registration file (LRF) for the SNMP managed object. Setting up this LRF file is one of the most important steps in being able to access SNMP managed objects with the CMIP services. The LRF file will convey the following information to the PostMaster[3]:

   Management program name This required field is not used by PostMaster SNMP access. But since it is required field, a dummy name needs to be supplied. For the sample LRF file shown in 14.3, "Sample LRF File" on page 211, we chose the name of **snmpAgent**.

   Agent executable file name This required field is not used by PostMaster for SNMP access. But since it is required field for all LRFs, a dummy name needs to be supplied. For the sample LRF file shown in 14.3, "Sample LRF File" on page 211, we chose the name of **SNMPD**.

   Quality of service This field is set to **OVs_SNMP_UDP** since SNMP uses the User Datagram Protocol (UDP) as the lower-level transport layer.

1 The standard MIB-2 has already been translated and integrated for you. So this only applies to MIB-2 extensions.

2 See Chapter 32 of the LAN NetView Manage 1.0 Administration Guide for an explanation of what is involved in translating the MIB.

3 The component of the LAN NetView communications infrastructure that routes all communication between management programs.

Object Class The registration ids of the SNMP managed object classes which may be accessed. For MIB-2, there are a total of 20 managed object classes that may be accessed. See the sample LRF file supplied in 14.3, "Sample LRF File" on page 211.

Object Instance The Distinguished Name (DN) for the object instance consists of one Relative Distinguished Name (RDN). The RDN is for the **cmotSystemId** and is set to the **IP** address of the target host.

4. Register the LRF file via the **SVADDOBJ** command. Be sure to specify the target host name whose SNMP MIB attributes are to be retrieved. For example, if the LRF file is named **lnsnmp.lrf** and the SNMP query is to be directed to a host whose name is **testMachine**, then the LRF file would be registered as follows:

   SVADDOBJ lnsnmp.lrf testMachine

5. Restart the PostMaster on the machine where the management application will be executed. This can be a machine with either LAN NetView Manage 1.0 or LAN NetView Enabler installed.

6. Design and code the management application program using the CMIP services:

   - A **CMIS-Get-Argument** OM class is used to specify the SNMP attributes that are to be retrieved from the target host.

     — The base managed object class is setup using an **Object-Class** OM class. If the SNMP attribute(s) to be retrieved are defined in MIB-2, then the designer may export the appropriate OM class using the labels defined in **MIB2.gdm**. For example, **MIB2_O_TCP_CONN_TABLE**.

     — The base managed object instance is setup using an **Object-Instance** OM class. This OM class would be intialized with the distinguished name of the target host. The distinguished name consists of one **RDN**. The **RDN** is made up of one **AVA** (Attribute Value Assertion) with the following OM attributes:

       - Attribute Id - The attribute id for **cmotSystemId**.
       - Attribute Value - The **IP** address of the target host.

     — Specify the optional Scoping level.

       - If the target attributes correspond to an aggregate table, specify a Scoping Level using the **Scope** OM class. The scoping level should be set to a **base-To-Nth-Level** of value 1.

       - If the target attributes do not correspond to an aggregate table, but instead correspond to scalar objects, no scoping is necessary.

     — Specify the optional attribute id list. If none are specified, all the attributes corresponding to the base managed object class will be retrieved. Use the **Attribute-Id-List** OM class if specifying an attribute id list.

   - Issue the **mp_get_req** function call to request the values specified in the attribute id list of the **CMIS_GET_ARGUMENT** OM class.

- If the **mp_get_req** function call returns a successful return code (that is,**MP_SUCCESS**), a public copy of the returned object is generated via the **om_get** function call.
- A **CMIS-Get-Result** OM class is used to parse the results and retrieve the desired attribute values.

14.3 Sample LRF File

This section contains a sample local registration file (LRF) for all of the managed object classes defined for MIB-2.

```
#
# Sample SNMP Agent Registration File
#
snmpAgent:SNMPD.EXE:OVs_SNMP_UDP:
OVs_NO_START:::OVs_NON_WELL_BEHAVED::
1.3.6.1.2.1.1:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.2:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.2.2:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.3:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.3.1:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.4:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.4.20:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.4.21:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.4.22:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.5:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.6:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.6.13:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.7:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.7.5:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.8:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.8.5:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.9:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.10:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.11:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
1.3.6.1.2.1.12:1.3.6.1.2.1.9.3=129.35.65.7:OVs_GETR::OVs_Pwd=MySecret:
```

Figure 12. Sample LRF File for MIB-2 Translation

14.4 Sample Code to Access tcpConnTable

This section contains a sample program that utilizes the CMIP services to retrieve all of the rows of a **TcpConnTable** aggregate table.

The output data is shown in 14.5, "Output of Sample Program" on page 219.

Note how a single **mp_get_req** invocation will return all of the attribute values for all of the rows of the target table. If the sample program had used the SNMP services, it would have required X number of invocations of the **mp_get_next_req** function call where X is the number of rows in the table.

```
#define INCL_DOSMODULEMGR
#include <os2.h>

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <xom.h>
#include <xmp.h>
#include <xmp_snmp.h>
#include <xmp_cmis.h>
#include <MIB2.h>
#include <lnv.h>


OM_EXPORT(OM_C_EXTERNAL);

OM_EXPORT(MP_C_AVA);
OM_EXPORT(MP_C_DS_DN);
OM_EXPORT(MP_C_DS_RDN);
OM_EXPORT(MP_C_SCOPE);
/* CMIS Package */

OM_EXPORT(MP_C_CMIS_GET_ARGUMENT);
OM_EXPORT(MP_C_OBJECT_CLASS);
OM_EXPORT(MP_C_OBJECT_INSTANCE);
OM_EXPORT(C_LNV_CMOT_SYSTEM_ID);
OM_EXPORT(LNV_A_CMOT_SYSTEM_ID);

OM_EXPORT(MIB2_O_TCP_CONN_TABLE);

/* static structures for CMIS public object get argument */

static OM_descriptor    public_cmot[] = {
   OM_OID_DESC(OM_CLASS, C_LNV_CMOT_SYSTEM_ID),
   {LNV_INET_ADDR, OM_S_OCTET_STRING, {4,"\x81\x23\x41\x07"}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_ava0[] = {
   OM_OID_DESC(OM_CLASS, MP_C_AVA),
   OM_OID_DESC(MP_NAMING_ATTRIBUTE_ID,LNV_A_CMOT_SYSTEM_ID),
   {MP_NAMING_ATTRIBUTE_VALUE, OM_S_OBJECT, {0, public_cmot} },
   OM_NULL_DESCRIPTOR
};
```

Figure 13 (Part 1 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
static OM_descriptor    public_rdn0[] = {
   OM_OID_DESC(OM_CLASS, MP_C_DS_RDN),
   {MP_AVAS, OM_S_OBJECT, {0, public_ava0}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_dn[] = {
   OM_OID_DESC(OM_CLASS, MP_C_DS_DN),
   {MP_RDNS,OM_S_OBJECT,{0, public_rdn0}},
   OM_NULL_DESCRIPTOR
};


static OM_descriptor    public_obj_inst[] = {
   OM_OID_DESC(OM_CLASS, MP_C_OBJECT_INSTANCE),
   {MP_DISTINGUISHED_NAME,OM_S_OBJECT,{0,public_dn}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_obj_class[] = {
   OM_OID_DESC(OM_CLASS, MP_C_OBJECT_CLASS),
   OM_OID_DESC(MP_GLOBAL_FORM,MIB2_O_TCP_CONN_TABLE),
   OM_NULL_DESCRIPTOR
};

static OM_descriptor    pub_scope[] = {
   OM_OID_DESC(OM_CLASS, MP_C_SCOPE),
   {MP_BASE_TO_NTH_LEVEL, OM_S_INTEGER, 1},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_get_arg[] = {
  OM_OID_DESC(OM_CLASS,MP_C_CMIS_GET_ARGUMENT),
  {MP_BASE_MANAGED_OBJECT_CLASS,OM_S_OBJECT, {0,public_obj_class}},
  {MP_BASE_MANAGED_OBJECT_INSTANCE,OM_S_OBJECT, {0,public_obj_inst}},
  {MP_SCOPE,OM_S_OBJECT,{0,pub_scope}},
  OM_NULL_DESCRIPTOR
};


  OM_workspace        ws;
  OM_return_code      omstat;
  MP_status         mp_ret;          /* MP return code            */
  OM_private_object   bound_session;
  OM_private_object   result;
  int              invoke;
  char              *temp_str;
  char              *input_str;
  int              i,rc;
  char              pszObjNameBuf[100];
  int              ulObjNameBufL;
  HMODULE              pModHandle;
   char              error_text[100]; /* error text for mp error*/

/**************************************************************************/
/* function:  display_network_address                                   */
/**************************************************************************/

void display_network_address (OM_string p)
{
  int i = 0;
  unsigned char * a = (unsigned char *) p.elements;
  for (i = 0; i < 3 ; i++)
  {
    printf ("%d.", *a++);
  };
  printf ("%d\n", *a);
}
```

Figure 13 (Part 2 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
/******************************************************************************/
/* function: display_oid_label                                               */
/******************************************************************************/

void display_oid_label (OM_object_identifier * oid,  OM_workspace ws  )
{
  char          *pkg_name_return;
  char          *label_name_return;
  OM_uint32     label_type_return;
  OM_return_code at_ret;

  at_ret = at_oid_to_label (ws, oid, &pkg_name_return,
                            &label_name_return,
                            &label_type_return );

  if (at_ret != OM_SUCCESS)
  {
    printf("Conversion failed with return code = %d\n",at_ret);
    exit(1);
  }

  printf ("%s = ",label_name_return);
  at_free (label_name_return);
  at_free (pkg_name_return);
}

/******************************************************************************/
/* function: display_object_identifier                                       */
/******************************************************************************/

void display_object_identifier (OM_string * s)
{
  char *p;
  at_oid_to_str (*s, &p);
  printf ("%s\n",  p);
  at_free (p);
}
```

Figure 13 (Part 3 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
/*****************************************************************************/
/* function         display_octet_string                                    */
/*****************************************************************************/

void display_octet_string (OM_public_object p)
  {
  int i = 0;
  unsigned char * a = (unsigned char *) p[1].value.string.elements;
  for (i = 0; i < 4 ; i++)
    printf ("%d.", *a++);
  printf ("\n");
  }


/*****************************************************************************/
/* function         display_any                                             */
/*****************************************************************************/

void display_any (OM_public_object pubObject ,OM_workspace ws )
{
  int i = 0;
  OM_public_object px;
  char  str[512];
  char        *pkg_name_return;
  char        *label_name_return;
  OM_uint32     label_type_return;
  OM_return_code at_ret;

  switch( pubObject[i].syntax & OM_S_SYNTAX ) {

    case OM_S_OBJECT:

      px = pubObject[i].value.object.object;

      at_ret = at_oid_to_label (ws, &(px[0].value.string), &pkg_name_return,
                          &label_name_return,
                          &label_type_return );
      if (at_ret == OM_SUCCESS) {
        at_free(label_name_return);
        at_free(pkg_name_return);
      } else{
        printf("Name unknown\n");
      } /* endif */

      for (i=1;px[i].type != OM_NO_MORE_TYPES ;i++ ) {
        at_type_to_label(ws,&(px[0].value.string),px[i].type,&label_name_return);

        display_any (&(px[i]) , ws);
        at_free(label_name_return);
        at_free(pkg_name_return);
      } /* endfor */
      break;
```

Figure 13 (Part 4 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
    case OM_S_INTEGER:

      printf( "%d \n",pubObject[i].value.integer);
      break;

    case OM_S_ENUMERATION:

      printf( "%3d\n", pubObject[i].value.enumeration );
      break;

    case OM_S_BOOLEAN:
    case OM_S_NULL:
      break;

    case OM_S_OBJECT_IDENTIFIER_STRING:

      display_object_identifier (&pubObject[i].value.string);
      break;

    case OM_S_BIT_STRING:
      break;
    case OM_S_OCTET_STRING:
      display_network_address (pubObject[i].value.string);
      break;
    case OM_S_GRAPHIC_STRING:
    case OM_S_PRINTABLE_STRING:
    case OM_S_ENCODING_STRING:
    case OM_S_GENERAL_STRING:
    case OM_S_GENERALISED_TIME_STRING:
    case OM_S_IA5_STRING:
    case OM_S_NUMERIC_STRING:
    case OM_S_VISIBLE_STRING:
    case OM_S_VIDEOTEX_STRING:
    case OM_S_UTC_TIME_STRING:
    case OM_S_OBJECT_DESCRIPTOR_STRING:
    memcpy (str ,
          pubObject[i].value.string.elements,
          pubObject[i].value.string.length );
    str[pubObject[i].value.string.length] = '\0' ; // add NULL
    printf("%s\n",&str[0]);
      break;

    default:

      break;
  }
}
```

Figure 13 (Part 5 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
/*****************************************************************************/
/* function        display_attribute_id                                       */
/*****************************************************************************/

void display_attribute_id (OM_public_object p, OM_workspace ws)
{
  int i = 0;
  while (p[i].type != OM_NO_MORE_TYPES)
  {
    switch (p[i].type)
    {
      case MP_GLOBAL_FORM:
        display_oid_label ( ( OM_object_identifier *)&p[i].value.string , ws);
        break;
      case MP_LOCAL_FORM:
        printf ("%d\n", p[1].value.integer);
      default:
        break;
    }
    i++;
  }
}

/*****************************************************************************/
/* function        display_attribute                                          */
/*****************************************************************************/

void display_attribute ( OM_public_object p, OM_workspace ws)
{
  int i = 0;
  OM_public_object px;
  while (p[i].type != OM_NO_MORE_TYPES)
  {
    switch (p[i].type)
    {
      case MP_ATTRIBUTE_ID:
        display_attribute_id (p[i].value.object.object, ws);
        break;
      case MP_ATTRIBUTE_VALUE:
        display_any (&p[i] , ws);
        break;
      default:
        break;
    }
    i++;
  }
}
```

Figure 13 (Part 6 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
/******************************************************************************/
/* function         display_get_result                                        */
/******************************************************************************/

void display_get_result (OM_public_object p, OM_workspace ws)
{
  int i = 0;
  while (p[i].type != OM_NO_MORE_TYPES)
  {
    switch (p[i].type)
    {
      case MP_ATTRIBUTE_LIST:
        display_attribute ( p[i].value.object.object,ws);
        break;
      default:
        break;
    }
    i++;
  }
}
/******************************************************************************/
/* function         display_cmis_linked_replies                               */
/******************************************************************************/

void display_cmis_linked_replies (OM_public_object p, OM_workspace ws)
{
  int i = 0;
  while (p[i].type != OM_NO_MORE_TYPES)
  {
    switch (p[i].type)
    {
      case MP_GET_RESULT:
        display_get_result (p[i].value.object.object, ws);
        printf("\n\n");
        break;
      default:
        break;
    }
    i++;
  }
}

/******************************************************************************/
/* function:          display_results                                         */
/*                                                                            */
/*   This function opens an CMIS GET ARGUMENT and displays its contents      */
/******************************************************************************/

void display_results (OM_private_object arg , OM_workspace ws)
{
  OM_public_object pub;
  OM_value_position tot;
  int i = 0;
  int count = 0;
  om_get (arg, OM_NO_EXCLUSIONS, 0, OM_TRUE, 0, 0, &pub, &tot);
  while (tot-- > 0)
  {
    switch (pub[i].type)
    {
      case MP_REPLIES:
        display_cmis_linked_replies (pub[i].value.object.object, ws);
        break;
      default:
        break;
    }
    i++;
  }
  om_delete (pub);
}
```

Figure 13 (Part 7 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

```
int  main(int argc, char ** argv){

  ws = mp_initialize();
  if (!ws){
    printf("initialize failure\n");
    exit(0);
  }


  at_activate_all_packages(ws);

  mp_ret = mp_bind(MP_DEFAULT_SESSION,ws,&bound_session);
  if (mp_ret != MP_SUCCESS){
    printf("bind failure\n");
    exit(0);
  }

  mp_ret = mp_get_req(bound_session,MP_DEFAULT_CONTEXT,pub_get_arg,&result,&invoke);

  if (mp_ret != MP_SUCCESS){
     mp_error_message(mp_ret,100,error_text);
     printf("mp_get_req : %s\n",error_text);

    exit(0);
  }

  display_results (result, ws);

  mp_ret = mp_unbind(bound_session);
  if (mp_ret != MP_SUCCESS){
    printf("unbind failure\n");
    exit(0);
  }

  mp_ret = mp_shutdown(ws);
  if (mp_ret != MP_SUCCESS){
    printf("shutdown failure\n");
    exit(0);
  }

  exit(0);
}
```

Figure 13 (Part 8 of 8). Sample Program to Retrieve tcpConnTable Using CMIP Services

14.5 Output of Sample Program

This section contains the output generated by executing the sample program found in 14.4, "Sample Code to Access tcpConnTable" on page 211. table.

```
Tcp-Conn-State = 2
Tcp-Conn-Local-Address = 0.0.0.0
Tcp-Conn-Local-Port = 21
Tcp-Conn-Rem-Address = 0.0.0.0
Tcp-Conn-Rem-Port = 0


Tcp-Conn-State = 2
Tcp-Conn-Local-Address = 0.0.0.0
Tcp-Conn-Local-Port = 23
Tcp-Conn-Rem-Address = 0.0.0.0
Tcp-Conn-Rem-Port = 0


Tcp-Conn-State = 2
Tcp-Conn-Local-Address = 0.0.0.0
Tcp-Conn-Local-Port = 1024
Tcp-Conn-Rem-Address = 0.0.0.0
Tcp-Conn-Rem-Port = 0


Tcp-Conn-State = 5
Tcp-Conn-Local-Address = 129.35.65.7
Tcp-Conn-Local-Port = 1026
Tcp-Conn-Rem-Address = 129.35.64.231
Tcp-Conn-Rem-Port = 23
```

Figure 14. Output from Sample Program

Chapter 15. Providing SNMP Management Using IBM LAN NetView

This paper describes the benefits of using a framework with an XMP API for creating management applications for SNMP agents. It includes coding reasons why the use of the XMP is preferable to using one of the SNMP interfaces available. Portability and portability considerations are presented as a benefit to using the industry standard interface. Protocol independence is explained as a further benefit. A discussion of the SNMP MIB function, the MIB extensions that are provided with LAN NetView (RMON, etc.), and how to add additional MIB extensions is also included, as well as a description of the Request Manager, and how it can be used for MIB browsing.

15.1 Introduction

One of the features of the LAN NetView product that will benefit customers and developers is its ability to support both the SNMP (Simple Network Management Protocol) and CMIP (Common Management Information Protocol) protocols. This major capability of the LAN NetView product highlights its comprehensiveness as a systems management platform. SNMP is a standard network management protocol that is supported by a vast majority of network devices like routers, hubs, concentrators, bridges, printers, etc. Any software or hardware can be managed in this way if it has an SNMP agent that represents managed objects contained in a supported Management Information Base (MIB).

An SNMP managed object is a specific type or class of management information (for example, a system description or an interface status). An instance of an SNMP object has a value that can be retrieved or set. Some objects have only a single instance for a given agent system (for example, system description). Other objects have multiple instances for a given agent system (for example, interface status for each interface on the system).

There are no SNMP agents shipped with the LAN NetView product; it is the responsibility of each hardware or software vendor to do so if it wishes its product to be managed in this way. But LAN NetView services can discover SNMP agents, construct a topology data base that a managing application can access, and, through its View component, present a GUI (Graphical User Interface) that shows the topology for selecting an object (like a machine or a software system) and launching an application to manage that object.

15.2 Understanding MIBs

A MIB is not necessarily a physically distinct database, but rather a collection of managed-object definitions. The SNMP and the OSI (Open Systems Interconnection) environments implement the MIB using different Structures of Management Information (SMI), different MIB definitions, and different naming conventions. Both environments register their MIB objects in a hierarchical tree structure defined by the International Standards Organization (ISO).

15.2.1 How MIBs are Organized

Conceptually, the SNMP MIB objects are organized in a hierarchical tree structure. Each branch in the tree has a unique name and numeric identifier (see Figure 15). Intermediate branches of the tree serve as a way to group related MIB objects together. The leaves of the tree represent the actual MIB objects. A subtree is used to refer to the entire group of branches and leaves under a particular intermediate branch. Figure 15 illustrates the tree structure.

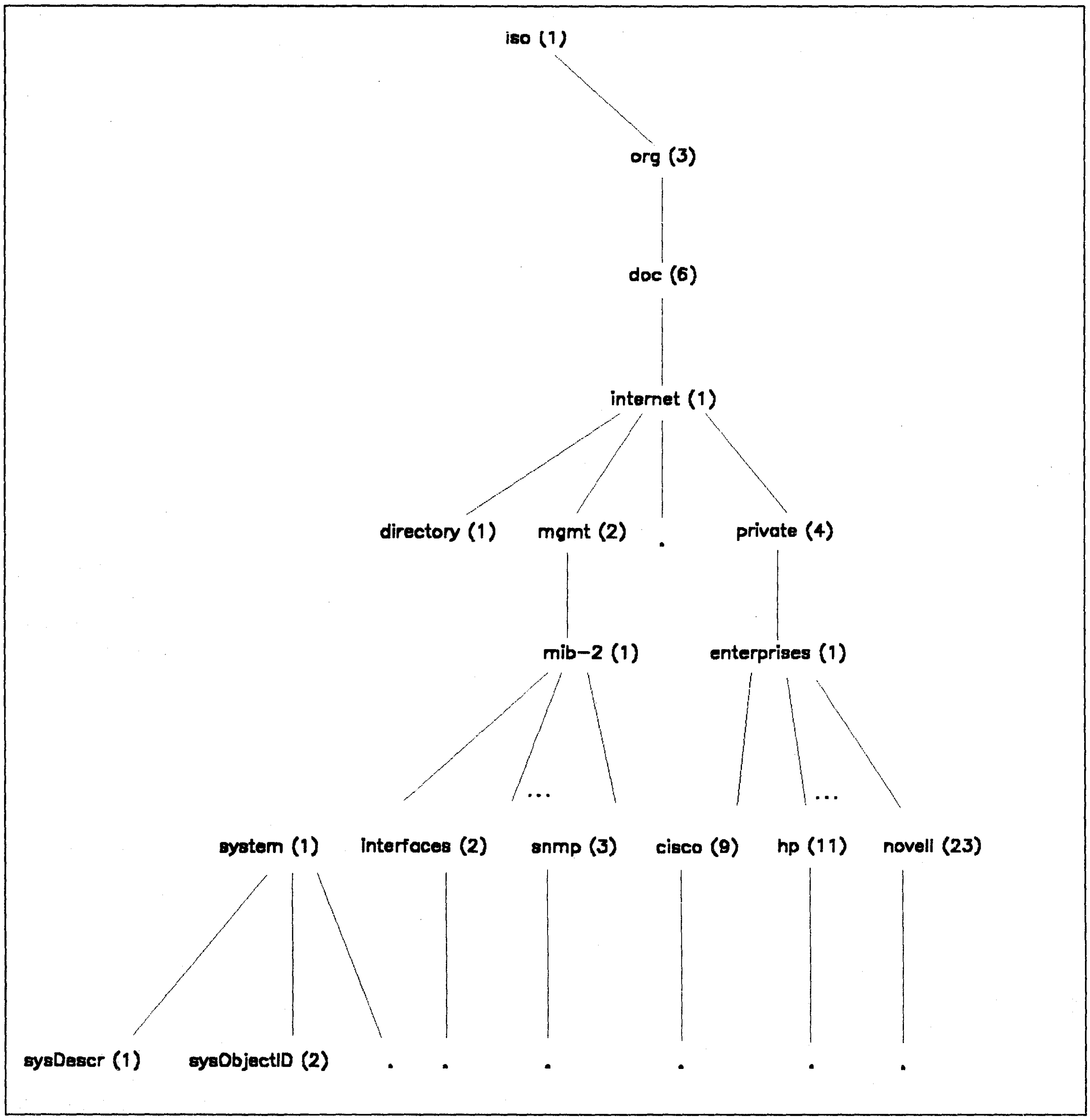


Figure 15. MIB Organization

An SNMP MIB object is uniquely identified (named) by its place in the tree. A full object identifier consists of the identifier of each branch along the path through the tree hierarchy, from the top of the tree (`iso`) down to the leaf (for example, `sysObjectID`). The object identifier is expressed in dotted notation by separating each branch identifier along the path with a period.

For example, the mib-2 subtree is `iso.org.dod.internet.mgmt.mib-2` and its numeric identifier is `1.3.6.1.2.1`.

As another example, the full MIB object identifier for `sysObjectID` is `iso.org.dod.internet.mgmt.mib-2.system.sysObjectID` and its numeric identifier is `1.3.6.1.2.1.1.2`.

The instance identifier on an SNMP MIB object with more than one instance is zero. The instance identifier on an SNMP MIB object with only one instance is one or greater. These MIB object notations follow the standard notation defined in Abstract Syntax Notation One (ASN.1); the ASN.1 standard notation definition can be considered the template for MIBs.

15.2.2 MIB Registration

To avoid conflicts of object IDs, each branch of the tree must be registered (that is, defined) through a designated organization. For example, the Internet Activities Board (IAB) has authority over the *internet* subtree. This includes the MIB-II Internet standard registered under the *mib-2* subtree.

The IAB then gives vendors authority over enterprise-specific subtrees. Enterprise-specific MIB objects are registered under the designated authority for that enterprise.

15.2.3 Using the MIB

Many managing applications can manage particular devices by hardcoding the MIB into the application. For more generic applications, LAN NetView Manage Version 1.0 explicitly supports MIB-II in its metadata base. MIB-II is a standard well-accepted Management Information Base for TCP/IP. The LAN NetView product supports MIB extensions for the IBM 6611 Multiprotocol Router and the SNMP RMON (Remote MONitoring) MIBs, and in the near future, intends to support many of the popular MIB extensions.

The MIB maintained in the metadata base is generated by a compiler from object descriptions stated according to the ISO Guidelines for Development of Managed Objects (GDMO). Any user additions to it would be described in GDMO format. This MIB is therefore called the GDMO metadata base. SNMP MIBs are first translated into GDMO before compilation into the metadata base.

Security is provided by password, or by limiting the permissible IP addresses of the requesters.

15.3 The XMP Interface

Support at the managing site is provided through the X/OPEN Management Protocol (XMP) interface, which contains a set of SNMP services and a set of CMIP services. XMP is a standard interface that allows either SNMP or CMIP to be supported underneath.

15.4 Benefits of using XMP

When a managing application uses the CMIP services of the XMP interface, it generally does not have to worry about whether the underlying management protocol layer is CMIP or SNMP, or whether the transport for CMIP is LAN (CMOL) or TCP/IP (CMOT). Thus a large degree of transparency to the underlying protocols is achieved.

However, in order to allow an application to take a CMIP view and still manage an SNMP device, a mapper from CMIP objects to SNMP objects is used. This mapper gets its metadata information from the Mapper Object Definition Interface (ODI) file (also supplied with LAN NetView Manage Version 1.0). This file is a Management Information Base containing MIB-II. But it can be extended by the user to contain any MIB extension. In order to add a published vendor MIB, "translation files" containing object descriptions are first prepared from the vendor published description known as Structure of Management Information (SMI). These translation files are processed by the SNMPMIB command. The translation files are considerably simpler than the full GDMO format. However, automatic compilation directly from SMI remains a task for the future.

15.5 What can be Managed with SNMP?

SNMP uses get, set, getnext and trap operations, which allow for the retrieval (get) and modification (set) of values like owner's name, traffic statistics, modem speed, etc. Each attribute is defined as a separate object in SNMP, and getnext is used to access subobjects of an agent object. Trap is used to generate events to a managing application.

Managing applications can be notified of unexpected events through the SNMP trap operation, supported by the Event Services component of XMP. Event Services manages both SNMP traps and ISO alarms. For example, if the FIX application receives an SNMP trap in this manner, it can display it at a console, log it, activate a beeper, retransmit it to another managing application, or take a user defined action.

A managing application can register for an event, and can supply filters to discriminate the particular events it wants to receive. Filtering can be done at the managed site or at the managing site, although most current SNMP agents do not support local filtering.

In addition, XMP provides convenience routines to encode to ASN.1, the general purpose syntax language used by various management protocols. This simplifies the work that has to be done by a managing application to manage a network.

15.6 How to Write an SNMP Manager

If you wish to write an SNMP managing application, you must first decide whether to use the CMIP or the SNMP features of the XMP interface. Either one can be used to manage an SNMP device. If the managing application treats the managed object as an SNMP object, it is not necessary to provide a MIB extension to the GDMO compiler. But it may still be desirable to do so, to allow the Request Manager to directly access the SNMP objects.

You can gain more flexibility, and a better organization of the objects for the managing view, by using CMIP services. To do this, you must provide translation tables to the ODI compiler, and GDMO object descriptions to the GDMO compiler. Then the application can be coded in very general terms, by accessing the GDMO metadata base to consult the object definitions, and by using only the CMIP view for all managed objects, whether they are actually CMIP managed objects or SNMP managed objects.

15.7 Content Packages

A content package is a DLL (Dynamic Link Library) that defines the data transformations needed to communicate between managing and managed nodes. For example, it would perform integer byte reversal and other required data transformations. A content package is produced by a tool that uses input taken from the SMI vendor description of a MIB extension. The content packages for MIB-II, RMON and IBM6611 are provided with LAN NetView Version 1.0. A new SNMP MIB extension would require a new content package to be generated.

15.8 The Request Manager

The Request Manager is a generic managing application for end users. It can manage any SNMP device whose MIB is defined in the GDMO metadata base. The LAN NetView user interface will present a user with a view of the network that is derived from information in the LAN NetView topology data base. This data base is constructed by a variety of discovery mechanisms. The user can select a top level object (a system, machine or operating system) from this view. He can then launch the Request Manager, which requests from the topology data base information telling it if the agent for the selected object is an SNMP agent. Request Manager communicates with the agent at the selected object to find the existing managed sub-objects at that site. These are displayed; if one is selected the Request Manager will obtain its definition from the GDMO metadata base (more precisely, for performance reasons, from a prepared-in-advance internal set of files derived from this metadata base).

At this point the object can be managed - that is, its values, shown in a notebook form, can be retrieved or set. For example, using only the Request Manager GUI at the managing site, the user can change the baud rate of a modem, or put a printer offline.

Chapter 16. Exploiting LAN NetView Metadata Services for a Generic Management Application

The Metadata Services of the LAN NetView family of products when exploited properly can be used to develop and maintain a truly generic management application.

This paper highlights the important features of Metadata Services that support the development of generic management applications. This is augmented by the presentation of a hypothetical generic management application[4].

16.1 Scope

This paper is aimed primarily at application developers who plan to implement systems/network management solutions on the IBM LAN NetView platform. It explains what Metadata Services consists of and how a developer can use these services to produce an effective management solution; one which has the advantage of being truly generic. This paper assumes that the reader has a working knowledge of the X/Open OSI-Abstract-Data Manipulation (XOM) and X/Open Management Protocol (XMP) application programming interfaces, as used by the LAN NetView family of products. The reader should also be familiar with the object modeling paradigm offered by LAN NetView.

The following LAN NetView publications provide additional information:

- *LAN NetView Manage and Enabler Version 1.0 Developer's Guide*, S96F-8493
- *LAN NetView Manage and Enabler Version 1.0 Developer's Reference*, S96F-8494

16.2 Introduction

The LAN NetView family of products provides a framework and management programs to implement OS/2-based network management solutions. The LAN NetView framework utilizes industry-standard interfaces and protocols that allow an OS/2 system to manage heterogeneous systems in a LAN environment. An OS/2 system also may be managed by other systems that conform to the same industry standards.

Metadata Services allows developers to create and delete managed-object class definitions in the management information base (MIB) of a LAN NetView distributed system. The definitions are stored in a private data store maintained by the Metadata agent. This data store is called the Metadata Database.

Management application programs may subsequently retrieve the managed-object class definitions from the Metadata Database using the query facilities of XMP. Among the reasons why a management application program would want to query the Metadata database are the following:

[4] The hypothetical application was chosen solely for the purpose of showing how a specific application can be designed and written to be a generic application. It also provides a common ground for the discussion of how the various convenience routines supplied by Metadata Services may be utilized. To further illustrate some points, sample program excerpts are presented , however the actual implementation of the application is beyond the scope of this paper.

- To retrieve information not provided by the XMP content package and not accessible via XMP convenience routines.

  For example:

  - Behavior description text
  - Name Binding information
  - Object Class inheritance

- In support of a generic management application.

16.3 Management Applications

Management applications are written using the X/Open OSI-Abstract-Data Manipulation (XOM) and X/Open Management Protocol (XMP) application program interfaces. Application designers make use of several convenience routines provided by the Manager and Enabler program products of the LAN NetView family of products.

16.3.1 The Managed Object Paradigm

Each managed object is under the control of an agent. An agent may be responsible for one or more managed objects. The agent owner creates a GDMO (see MIB description below) definition describing the managed object. This description includes the object's attributes as well as any actions that a management application may initiate on it. The ASN.1 definition required to support the complete definition of a managed object is also the responsibility of the agent owner.

Management application programs subsequently retrieve and manipulate the managed objects by interacting with the agent via the XMP application program interfaces. At no point, would an application access the managed object directly. It is always via the agent.

The agent owner provides the following deliverables to management applications:

MIB An ASCII file that contains the full GDMO description of the managed object. The descriptions conforms to the following standard:

> "ISO 10165-4, Management Information Services-Structure of Management Information Part 4: Guidelines for the Definition of Managed Objects".

The ASN.1 description referenced by the GDMO definitions is also included in the MIB file.

DLL A dynamically linked module that contains the XMP management content package for the referenced managed object. This package effectively contains the encoding\decoding information that is required by XMP.

Header File The content package header file provides #defines (that is, labels) for class oids and class syntax information. These labels are used by management applications to create and to parse the OM data structures utilized in conjunction with the XMP application program interface.

LRF File The Local Registration File (LRF) is a specially formatted ASCII file that contains information about an agent. For example, its name, where its executable code is located, how to start it, and the managed objects ,if any, for which it is responsible.

16.3.2 Categorizing Management Applications

Most management applications may be grouped into the following categories:

specialized
generic

16.3.2.1 Specialized Management Applications

These applications are written for a predefined set of managed objects. This assumes that the application designer has access to the managed object's content package and to its header file. Hence, the application designer knows exactly what the interface to the managed object as presented by the agent owner is. The application designer assumes that the description of the managed object will not change over the course of time.

16.3.2.2 Generic Management Applications

As the name implies, a generic management application is written to handle more than one type of managed object. For each managed object, the definition is allowed to change over time without causing a change to the management application program itself. It other words, it can dynamically adapt to new managed object classes as well as to changes to the definitions of existing managed object classes[5].

As one would expect, generic management applications will more than likely be more complicated and more processor intensive than specialized management applications.

16.4 An Example of a Specialized Management Application

Before discussing how Metadata Services may be exploited in support of generic management applications, this section introduces a hypothetical specialized management application. In a subsequent section, we will discuss how the application may be enhanced so that it becomes generic in nature.

This hypothetical management application is designed for LAN NetView and as such will utilize the XOM and XMP application program interfaces.

16.4.1 Description

Our hypothetical application, PrintMaster, manages a pre-defined set of printers connected on a network.

The set currently includes only laser printers.

[5] If should be pointed out that a properly designed managed object would not likely change its definition to avoid the chaos that would be associated with such a change. For example a specification once it becomes a standard tends to avoid definition changes.

16.4.2 Functional Requirements

The following list describes the requirements for the PrintMaster application:

1. Distributed application

   The printers may be located anywhere on the network.

2. The application should provide the following information for each of the known printers:

   - Current queue status
   - Paper Bin status
   - Toner supply status

3. Several user initiated actions are to be supported. These include

   - Cancel a print job
   - Stop a print job
   - Submit a print job

16.4.3 Modeling the Application

Our sample application is modelled using the managed object classes described in this section. A snippet of the MIB is provided in 16.8, "PrintMaster MIB File" on page 245.

16.4.3.1 Printer

Since all printers have several things in common, the managed object design paradigm leads to the creation of a super class. The **Printer** managed object class offers the following attributes:

- Name
- Type
- Job queue
- Status

Actions permitted on this managed object class might include:

- Cancel job
- Stop
- Resume
- Submit job
- Display queue

The distinguishing attribute for the **Printer** managed object class would be the printer name.

16.4.3.2 LaserPrinter

The **LaserPrinter** managed object class models the laser printers currently supported by PrintMaster.

The **LaserPrinter** managed object class is inherited from the **Printer** managed object class and as such inherits all of the former's attributes and actions. However it also defines those attributes\actions unique to a laser printer. For example:

- Toner status
- Paper Bin status

The distinguishing attribute for the LaserPrinter managed object class would be the printer name.

16.4.4 Creating the Content Package Deliverables

As previously discussed in 16.3.1, "The Managed Object Paradigm" on page 228, the agent owner must provide the deliverables that comprise the content package.

A typical logic flow follows:

1. The GDMO definition MIB is written using the guidelines outlined by ***ISO 1065-4***.
2. The MIB's syntax is verified using the syntax check features of LAN NetView's Metadata Compiler (METACOMP).
3. At this point, the agent owner makes the decision whether or not to "compile" the MIB into the Metadata database. This is an optional step given that PrintMaster is a specialized management application. The PrintMaster application can be written independent of this step. Compiling the MIB allows access to the managed object's definition by third parties. This will be addressed later in this paper.
4. The XMP content package header file is created.

   As mentioned previously, this header file contains defines that facilitate the definition of the OM data structures passed on as input\output arguments in various XMP application program interfaces.

   A XMP content package header file contains the following:

   - The BER-encoded values for the object identifiers for all the OM object definitions:
     - Managed Objects
     - Attributes
     - Attribute Groups
     - Notifications
     - Actions
     - Parameters
     - Name Bindings
     - Packages
     - Class Objects
   - Attribute Name Constants
   - Enumerated Value Lists

   For an example of an XMP content package header file, please refer to the META.H file delivered with the LAN NetView Manage product.

   A snippet of the XMP content package for the PrintMaster application is found in 16.9, "Content Package Header File" on page 248.
5. The XMP content package source file is created.

   The content package frees an XMP application from having to provide its own encoding\decoding routines. In this scenario, XMP encodes\decodes the OM objects for the application.

   The XMP content package source file is a "c" source file that defines the OM classes that correspond to the GDMO templates and ASN.1 syntaxes found in the MIB file. The source file defines the OM data structures using the OM template data structures defined in "xom.h" and "pkg_hdr.h".

The OM_EXPORT macro is used to create variables which are initialized with values defined in the content package header file. These exported variables are then used to initialize the OM data structures.

16.10, "Content Package Source File" on page 249 contains a snippet of the content package source file for the PrintMaster application.

6. The XMP content package DLL file is created.

   The content package source file is compiled and linked to produce the content package DLL.

   The LAN NetView developer's kit provides a robust list of content packages. Please refer to Chapter 5 of the LAN NetView Manage and Enabler Developer's Reference. As an example, META.DLL is the content package for the Metadata agent.

16.5 Designing and implementing the Agent

Once the OM objects have been modelled and the content packages defined, the individual XMP agents are designed and implemented.

PrintMaster requires the following XMP agents:

- LaserPrinter agent

Each XMP agent is designed and written to manage its specific managed object class. A local registration file (LRF) is created for each agent. The LRF file , as has been previously stated, contains specific information for the agent. For example, its name, where its executable code is located, startup parameters, dependencies, and the managed objects, if any, for which it is responsible.

XMP agents perform the following:

- Process XMP messages
- Handle filtering and scoping
- Issue Events where appropriate
- Maintain the internal data store where appropriate

Each XMP agent is registered with the LAN NetView Object Registration Service (ORS) using the **SVADDOBJ** command.

Since PrintMaster is to be designed as a distributed application, each agent LRF is registered at all client nodes. This can be done in one of two ways. The LRF file can be registered manually or the Master\Slave feature of ORS can be utilized.

16.5.1 Designing and Implementing the Application

The PrintMaster application is now ready to be designed and implemented. It utilizes the services of XOM and XMP to query, retrieve, and initiate actions on PrintMaster managed objects..

PrintMaster, itself a LAN NetView XMP application, follows the design recommendations outlined in the *LAN NetView Manage and Enabler Developer's Guide.*

The various OM data structures which need to be passed as arguments in many of the XMP application program interfaces are initialized in a very straightforward fashion since the application has access to the various content package deliverables described earlier.

Processing of the results returned by the PrintMaster agents is also quite straightforward due to the availability of the various OM object type labels found in the content package header files.

The following section(s) illustrate how a typical application program sets up the OM data structures required on input to the various XMP function calls. Note how the application program does not need to concern itself with encoding\decoding the OM attributes.

16.5.1.1 Managed Object Class

The **Managed Object Class** OM attribute specifies the class of the managed object for which one or more attributes are to be read.

This OM attribute usually takes on the global form of an object identifier. For example, assume that the application is to retrieve one or more attributes from the **LaserPrinter** managed object class. The **Managed Object Class** OM attribute would be initialized as shown in Figure 16.

```
static OM_descriptor public_obj_class[] = {
   OM_OID_DESC(OM_CLASS, MP_C_OBJECT_CLASS),
   OM_OID_DESC(MP_GLOBAL_FORM,PX_O_LASER_PRINTER),
   OM_NULL_DESCRIPTOR
};
```

Figure 16. OM Descriptors for MP_C_OBJECT_CLASS

The variable PX_O_LASER_PRINTER is exported using the OM_EXPORT macro. OM_EXPORT creates a variable that is initialized with the value of the OMP_O_PX_O_LASER_PRINTER #define.

16.5.1.2 Managed Object Instance

The **Managed Object Instance** OM attribute specifies the instance of the base managed object for which one or more attributes are to be read. This OM attribute usually takes on the form a DS-DN OM attribute.

For example, assume a query is to be directed to the LaserPrinter named **Printer1**. The corresponding **Managed Object Instance** OM attribute is completed as shown in Figure 17 on page 234.

```
static OM_descriptor public_obj_inst[] = {
   OM_OID_DESC(OM_CLASS, MP_C_OBJECT_INSTANCE),
   {MP_DISTINGUISHED_NAME,OM_S_OBJECT,{0,public_dn}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_dn[] = {
   OM_OID_DESC(OM_CLASS, MP_C_DS_DN),
   {MP_RDNS,OM_S_OBJECT,{0,public_rdn0}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_rdn0[] = {
   OM_OID_DESC(OM_CLASS, MP_C_DS_RDN),
   {MP_AVAS,OM_S_OBJECT,{0,public_ava0}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_ava0[] = {
   OM_OID_DESC(OM_CLASS, MP_C_AVA),
   OM_OID_DESC(MP_NAMING_ATTRIBUTE_ID,PX_A_PRINTER_NAME),
   {MP_NAMING_ATTRIBUTE_VALUE,{{8,"Printer1"}}},
   OM_NULL_DESCRIPTOR
};
```

Figure 17. OM Descriptors for MP_C_OBJECT_INSTANCE

The variable PX_A_PRINTER_NAME is defined using the OM_EXPORT macro.

16.5.1.3 Attribute Id List

The **Attribute Id List** OM attribute specifies the list of identifiers that specify attributes for which a value is to be returned.

For example, assume the attribute values for toner status and paper bin status are to be retrieved. The corresponding OM object descriptors would be completed as shown in Figure 18 on page 235.

The defines PX_A_TONER_STATUS and PX_A_PAPER_BIN_STATUS are exported using the OM_EXPORT macro.

16.5.1.4 Attribute

The previous sections have shown how an application sets up the OM data structures for input to XMP calls. This section shows some examples of how the the OM objects that are returned on completion of an XMP call are parsed.

The CMIS-Get-Result is the result of successful CMIS get operation. Its OM descriptor is composed of the following OM attributes:

- class
- managed-Object-Class
- managed-Object-Instance
- current-Time
- attribute-List

The OM descriptor for **attribute-List** is composed of the following OM attributes:

- class

```
static OM_descriptor public_attr_list[] = {
   OM_OID_DESC(OM_CLASS, MP_C_ATTRTIBUTE_ID_LIST),
   {MP_ATTRIBUTE_IDS,OM_S_OBJECT,{0,public_attrId0}},
   {MP_ATTRIBUTE_IDS,OM_S_OBJECT,{0,public_attrId1}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_attrId0[] = {
   OM_OID_DESC(OM_CLASS, MP_C_ATTRIBUTE_ID),
   OM_OID_DESC(MP_GLOBAL_FORM,PX_A_TONER_STATUS),
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_attrId1[] = {
   OM_OID_DESC(OM_CLASS, MP_C_ATTRIBUTE_ID),
   OM_OID_DESC(MP_GLOBAL_FORM,PX_A_PAPER_BIN_STATUS),
   OM_NULL_DESCRIPTOR
};
```

Figure 18. OM Descriptors for MP_ATTRIBUTE_ID_LIST

- attribute-Id
- attribute-Value

Referring back to the example found in 16.5.1.3, "Attribute Id List" on page 234, let's assume that the application has successfully issued a **mp_get_req** call. After verifying that the return code was successful, a public version of the ***argument*** parameter is obtained using the **om_get** call.

The application must now parse through the public object searching for the **attribute-List** OM attribute. Once it is found, each for the **attribute-List** OM attribute. Once it is found, each OM attribute which comprises the attribute-List must be parsed and processed to retrieve the attribute values.

Before the attribute value is retrieved, the application must figure out which attribute the attribute value corresponds to. The program excerpt, found in Figure 19 on page 236, shows a typical logic design. Note that the program to parse through the **attribute-Value** is not shown for space considerations.

Figure 19 on page 236 shows how PrintMaster Version 1.0 made use of the PX_A_TONER_STATUS and PX_A_PAPER_BIN_STATUS exported labels to parse through the results returned from the **mp_get_req** call.

16.5.2 A Final Word on Specialized Management Applications

This section has attempted to demonstrate how most management applications written to the XMP application program interface utilize the agent's XMP content package(s).

The purpose of this paper is to show how Metadata Services may be exploited to support generic management applications. This can now be discussed.

```
void display_results (OM_private_object arg, OM_workspace ws)
{
  OM_public_object pub;
  OM_value_position tot;
  int i=0;
  int count = 0;
  OM_public_object &px;
  om_get (arg, OM_NO_EXCLUSIONS, 0 , OM_TRUE,0,0,&pub,&tot);
  while ( tot-- > 0 )
  {
     switch(p[i].type)
     {
        case MP_ATTRIBUTE_LIST:
              px = p[i].value.object.object;
              while (px->type != OM_NO_MORE_TYPES)
              {
                  switch (px->type)
                  {
                    case MP_ATTRIBUTE_ID:
                       if (px->value.string == PX_A_TONER_STATUS)
                       {
                         printf("Toner Status = ");
                       }
                       else
                       {
                          if (px->value.string == PX_A_PAPER_BIN_STATUS)
                          {
                            printf("Paper Bin Status = ");
                          }
                          else
                          {
                              printf("Unknown type found \n");
                              exit(-1);
                          }
                       }
                       break;
                    case MP_ATTRIBUTE_VALUE:
                       display_any (px,ws);
                       break;
                    default:
                       break;
                  }
                  px = px + 1;
              }
        default:
          break;
     }
     i = i + 1 ;
  }
```

Figure 19. Parsing for Attribute Value

16.6 An Example of a Generic Management Application

16.6.1 When is a Generic Management Application Needed?

The objective of a generic management application is to be able to handle a dynamic set of managed objects whose definitions are allowed to change.

For example, suppose a management application desires to handle all types of routers. The list of available routers is dynamic and it is a customer requirement that the application not have to be modified every time a new type of router is added to the system. In this situation, the application is better suited if it is designed and implemented as a generic management application.

It should come as no surprise that there is a price to pay for designing a generic management application. The application will be more complex, but the cost may be acceptable if the application meets its functional requirements.

This section will again make use of the PrintMaster application and expand on it to demonstrate how it can be redesigned as a generic management application and what features of Metadata Services should be exploited.

16.6.2 Changing the Functional Requirements of PrintMaster.

The following changes to the PrintMaster application are proposed to showcase it as a generic management application.

- Support all types of printers on the network.
- Dynamically adapt to model changes.

16.6.3 Effects on the PrintMaster Logical Model

The following functional requirements have been added:

- Support all types of printers.
- Dynamically adapt to model changes.

16.6.3.1 Support All Types of Printers

This functional enhancement is accomplished by the fact that all printers are derived from the **Printer** managed object class. To add a new type of printer, a corresponding managed object class needs to be created. For example, a ColorPrinter managed object class would be created if one or more color printers were to be added to the network.

16.6.3.2 Dynamically Adapt to Model Changes

This functional requirement implies a change[6] in the GDMO definition of an existing managed object class. For example, suppose the older laser printers were replaced by a newer version.

[6] Metadata Services does not support changes in the definition of a managed object class. The current definition must be deleted from the Metadata Database via the *-r* option in **METACOMP**. Once the deletion occurs, a new definition may be integrated once again using the **METACOMP** program.

16.6.4 Effects on the PrintMaster Agents

As with the logical model, new agents would need to be designed and implemented to handle the new printers. The corresponding content package deliverables would also need to be designed and implemented.

The goal of making PrintMaster Version 2.0 a generic management application is that the application would not need to change. The PrintMaster administrator would simply install the new agent and its content package deliverables. As previously mentioned,the Metadata Database would also be updated to reflect the new deliverables.

16.6.5 Effects on the PrintMaster Application

When it was designed as a specialized management application, the PrintMaster application could readily answer the following:

- What managed object classes to handle?
- What attributes can be queried?
- What syntax is associated with an attribute?
- What actions are available?
- What is the naming tree?

As a generic application, PrintMaster will now need to rely on Metadata Services to obtain the answers to most if not all of these questions. With the right answers, a generic managed application can then utilize XMP and XOM convenience routines to complete the picture.

The remaining sections will take each question and demonstrate how a modified PrintMaster application answers the question(s). The specialized PrintMaster application will be referred to as PrintMaster Version 1.0, whereas the generic PrintMaster application is referred to as PrintMaster Version 2.0.

16.6.5.1 What Managed Object Classes to handle?

PrintMaster Version 1.0 was designed and implemented to expect one kind of printer (for example, a laser printer). As was shown in 16.5.1.1, "Managed Object Class" on page 233, PrintMaster Version 1.0 used the PX_O_LASER_PRINTER exported variable to set up the Managed Object Class OM attribute of the CMIS_GET_ARGUMENT OM descriptor.

In order to handle all types of printers, PrintMaster Version 2.0 can no longer use these variables.

PrintMaster Version 2.0 makes use of the fact that every printer has been modelled such that it is derived from the **Printer** managed object class. So to answer the question "What managed object classes to handle?", PrintMaster Version 2.0 must now determine all the managed object classes that are derived from the **Printer** managed object class.

This may be accomplished by querying the Metadata Database. The Metadata Database has been modelled as a managed object, and as such it contains attributes which may be queried and actions that may be initiated on it.

Of particular importance to the discussion in this section is the **objectClass** managed object class. This managed object class has the following attributes:

registration Unique registration id for this managed object class.

templateName Unique name for this managed object class.

derivedFromList List of managed object classes that this managed object class derives attributes,actions,etc..

charByList List of mandatory packages for this managed object class.

condPkgList List of conditional pacakges for this managed object class.

Performing a scoped get with a filter allows PrintMaster Version 2.0 to retrieve from the Metadata Database the list of all managed object classes that are derived from the **Printer** managed object class.

The OM descriptors required to set up the CMIS_GET_ARGUMENT OM class are shown in Figure 20 on page 240.

Given the CMIS_GET_ARGUMENT OM attribute described in Figure 20 on page 240, the corresponding mp_get_req call would return an attribute list containing the following:

- registration id
- template name

So at this point, PrintMaster Version 2.0 has determined the names and registration ids corresponding to the new types of printers that it should handle.

16.6.5.2 What Attributes Can Be Queried?

Having determined the list of managed object classes that it should handle, PrintMaster Version 2.0 must now answer the question:

```
What attributes can be queried?
```

This information may be obtained in one of two ways:

- performing multiple queries to the Metadata Database
- initiating the **getAttrSructList** action

Multiple Queries to Obtain List of Attributes: Obtaining the list of attributes for a given managed object class cannot be accomplished with a single query given the relationship of GDMO attributes to GDMO managed object classes. Attributes are referenced by mandatory packages and conditional packages. The packages are in turn referenced by the managed object class. So in order to determine the list of attributes for a given managed object class, PrintMaster Version 2.0 must first determine the list of mandatory and conditional packages. Once having obtained this initial list, a subsequent query will yield the final list.

For mandatory packages, PrintMaster Version 2.0 will need to perform a query for each mandatory package in the list. Since mandatory packages are not required to have registration ids associated with them, the query would be structured to use the more complex name binding for the **package** managed object class:

```
\metaDataId=0\docName=xxx\templateName=yyy
  where xxx and yyy are obtained from the results of the query
  for the list of packages for a given managed object class.
```

For conditional packages, PrintMaster Version 2.0 may elect to perform the query using the simpler name binding for the **package** as follows:

```
static OM_descriptor  pub_get_arg[] = {
   OM_OID_DESC(OM_CLASS,MP_C_CMIS_GET_ARGUMENT),
   {MP_BASE_MANAGED_OBJECT_CLASS,OM_S_OBJECT, {0,public_obj_class}},
   {MP_BASE_MANAGED_OBJECT_INSTANCE,OM_S_OBJECT, {0,public_obj_inst}},
   {MP_ATTRIBUTE_ID_LIST,OM_S_OBJECT,{0,pub_attribute_id_list}},
   {MP_SCOPE,OM_S_OBJECT,{0,pub_scope}},
   {MP_FILTER,OM_S_OBJECT,{0,pub_filter}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_obj_class[] = {
    OM_OID_DESC(OM_CLASS, MP_C_OBJECT_CLASS),
    OM_OID_DESC(MP_GLOBAL_FORM, META_O_METADATA ),
    OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_obj_inst[] = {
    OM_OID_DESC(OM_CLASS, MP_C_OBJECT_INSTANCE),
    {MP_DISTINGUISHED_NAME,OM_S_OBJECT,{0,public_dn}},
    OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_dn[] = {
    OM_OID_DESC(OM_CLASS, MP_C_DS_DN),
    {MP_RDNS,OM_S_OBJECT,{0, public_rdn0}},
    OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_rdn0[] = {
    OM_OID_DESC(OM_CLASS, MP_C_DS_RDN),
    {MP_AVAS, OM_S_OBJECT, {0, public_ava0}},
    OM_NULL_DESCRIPTOR
};

static OM_descriptor    public_ava0[] = {
    OM_OID_DESC(OM_CLASS, MP_C_AVA),
    OM_OID_DESC(MP_NAMING_ATTRIBUTE_ID, META_A_META_DATA_ID ),
    {MP_NAMING_ATTRIBUTE_VALUE,OM_S_INTEGER, 0},
    OM_NULL_DESCRIPTOR
};


static OM_descriptor  pub_attribute_id_list[] = {
   OM_OID_DESC(OM_CLASS,MP_C_ATTRIBUTE_ID_LIST),
   {MP_ATTRIBUTE_IDS,OM_S_OBJECT, {0, pub_attr0}},
   {MP_ATTRIBUTE_IDS,OM_S_OBJECT, {0, pub_attr1}},
   OM_NULL_DESCRIPTOR
};
```

Figure 20 (Part 1 of 2). CMIS_GET_ARGUMENT for Scoped, Filtered Get

```
\optionalRegistration=x.y.z
   where x.y.z is the registration id returned by the query
   for the list of packages for a given managed object class.
```

Initiating the getAttrSructList Action: Metadata Services provides this

```
static OM_descriptor  pub_attr0[] = {
   OM_OID_DESC(OM_CLASS,MP_C_ATTRIBUTE_ID),
   OM_OID_DESC(MP_GLOBAL_FORM,META_A_REGISTRATION),
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_attr1[] = {
   OM_OID_DESC(OM_CLASS,MP_C_ATTRIBUTE_ID),
   OM_OID_DESC(MP_GLOBAL_FORM,META_A_TEMPLATE_NAME),
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_scope[] = {
   OM_OID_DESC(OM_CLASS,MP_C_SCOPE),
   {MP_INDIVIDUAL_LEVELS,OM_S_INTEGER, 2},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_filter[] = {
   OM_OID_DESC(OM_CLASS,MP_C_CMIS_FILTER),
   {MP_ITEM,OM_S_OBJECT, {0, pub_filter_item}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_filter_item[] = {
   OM_OID_DESC(OM_CLASS,MP_C_FILTER_ITEM),
   {MP_EQUALITY,OM_S_OBJECT, {0, pub_filter0}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_filter0_attribute_id[] = {
   OM_OID_DESC(OM_CLASS,MP_C_ATTRIBUTE_ID),
   OM_OID_DESC(MP_GLOBAL_FORM,META_A_DERIVED_FROM_LIST),
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_filter0_attribute_value[] = {
   OM_OID_DESC(OM_CLASS,C_META_REG_ID_LIST),
   OM_OID_DESC(META_REG_ID, PX_O_PRINTER ),
   OM_NULL_DESCRIPTOR
};

static OM_descriptor  pub_filter0[] = {
   OM_OID_DESC(OM_CLASS,MP_C_ATTRIBUTE),
   {MP_ATTRIBUTE_ID,OM_S_OBJECT,    {0, pub_filter0_attribute_id}},
   {MP_ATTRIBUTE_VALUE,OM_S_OBJECT, {0, pub_filter0_attribute_value}},
   OM_NULL_DESCRIPTOR
};
```

Figure 20 (Part 2 of 2). CMIS_GET_ARGUMENT for Scoped, Filtered Get

convenient action[7] as part of its **MetaStruct** managed object class. The action is directed against the **MetaStruct** managed object class. The name binding for this managed object class is

---

[7] See 16.7, "Actions Supported by Metadata Services" on page 244 for the complete list of actions provided by Metadata Services.

```
\dbId=0
```

As is the case with the **metaData** managed object class, there is only one instance of the **metaStruct** managed object class.

The input to the **getAttrStructList** action is the registration id of the managed object class whose attributes are to be retrieved. The following is returned for each attribute:

id	registration id for attribute
name	qualified name of attribute
matchesFor	settings for matches for clause
derivedFrom	qualified name of parent attribute if derivedFrom clause specified
withAttributeSyntax	qualified name of ASN.1 syntax for this attribute
behaviorList	list of behavior labels
parameterList	qualified name list of parameters associated with this attribute.

If designed properly, PrintMaster Version 2.0 will make use of the **behaviorList** attribute to provide the PrintMaster Version 2.0 end user with a textual description of each attribute.

After obtaining the registration ids for the attributes for a given managed object class, the MP_C_ATTRIBUTE_ID_LIST OM attribute can now be completed. This is now dynamically updated as opposed to statically defined as was shown in 16.5.1.3, "Attribute Id List" on page 234.

Let's assume that PrintMaster Version 2.0 has successfully gotten the list of attributes for a given managed object class. It has presented the list to the end user who in turn has selected two attributes whose values are to be retrieved. Furthermore,assume that the registration ids (in BER-encoded format) has been stored in the following two program variables:

```
OM_object_identifier  oid1,oid2;
```

Figure 21 on page 243 illustrates how the two registration ids can be used to setup the MP_C_ATTRIBUTE_ID_LIST OM attribute. Note that the logic to retrieve the registration ids for oid1 and oid2 is not shown.

16.6.5.3 What Syntax is Associated with an Attribute?

The next problem that arises for PrintMaster Version 2.0 is how to parse the results returned by the PrintMaster agent that responded to the mp_get_req request.

As was shown in Figure 19 on page 236, the exported variables (for example, PX_A_TONER_STATUS and PX_A_PAPER_BIN_STATUS) were used to determine what ATRIBUTE-ID,ATTRIBUTE-VALUE pair was being processed.

What wasn't shown was the logic that parses the ATTRIBUTE_VALUE OM attribute of the MP_C_ATTRIBUTE OM class. Since the syntax for attribute value is defined as **any**, special logic has to be added to parse for the corresponding value.

```
static OM_descriptor public_attr_list[] = {
   OM_OID_DESC(OM_CLASS, MP_C_ATTRTIBUTE_ID_LIST),
   {MP_ATTRIBUTE_IDS,OM_S_OBJECT,{0,public_attrId0}},
   {MP_ATTRIBUTE_IDS,OM_S_OBJECT,{0,public_attrId1}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_attrId0[] = {
   OM_OID_DESC(OM_CLASS, MP_C_ATTRIBUTE_ID),
   {MP_GLOBAL_FORM,OM_S_OBJECT_IDENTIFIER,{0,NULL}},
   OM_NULL_DESCRIPTOR
};

static OM_descriptor public_attrId1[] = {
   OM_OID_DESC(OM_CLASS, MP_C_ATTRIBUTE_ID),
   {MP_GLOBAL_FORM,OM_S_OBJECT_IDENTIFIER,{0,NULL}},
   OM_NULL_DESCRIPTOR
};
              .
              .
              .
public_attrId0[1].value.string = oid1;
public_attrId1[1].value.string = oid2;
              .
              .
```

Figure 21. OM Descriptors for MP_ATTRIBUTE_ID_LIST

Once the ATTRIBUTE-ID OM attribute has been parsed, PrintMaster Version 2.0 can then determine the ASN.1 syntax for the attribute and thus have the necessary information to process the **any** syntax.

The ASN.1 syntax for a given attribute can be obtained in several ways. For example:

- performing multiple queries to the Metadata Database.
- initiating the **getSyntaxStruct** action
- utilizing the **at_oid_to_syntax** XMP convenience routine

The first two approaches require considerable setup and processing. By far the recommended approach is the use of the **at_oid_to_syntax** XMP convenience routine. Hence we will limit our discussion to this approach only.

The **at_oid_to_syntax** routine returns the necessary OM_syntax values that will assist in the parsing of the attribute's value. This convenience routine is designed to process even complex ASN.1 types whose OM_syntax equals OM_S_OBJECT.

Note: At this point, it may be beneficial to pause for an instant and clarify an important assumption that permeates this paper. The intent of this paper has been to demonstrate the benefits of using Metadata Services in generating generic management applications. We have stated that it is possible to create a generic managed application that can handle a dynamic list of managed object classes whose definitions are allowed to change. We have illustrated this with the ongoing hypothetical application PrintMaster showing how it must be modified. But perhaps we have not stressed enough the fact that unless the agent owner provides the appropriate content package deliverables, XMP may be unable to free the application from having to worry about encoding\decoding. Without the proper content package, XMP will simply return the result as an ***any*** structure if the attribute being processed is complex. For these cases, XMP will default the decoding of the attribute to the application.

16.6.5.4 What Actions are Available?

Unfortunately, Metadata Services does not provide an action analogous to **getAttrStructList** to retrieve the actions defined for a managed object class. However, the list of actions can nevertheless be obtained by querying the **package** managed object class for the **actionList** parameter.

As with the list of attributes, PrintMaster Version 2.0 should take advantage of the behavior text associated with each action to provide a textual description as to what exactly the purpose of the action is.

Having gotten the registration id for the actions for a given managed object class, the XMP convenience routine , **at_oid_to_syntax** may be called to retrieve the action's information and reply syntax information.

16.6.5.5 What is the Naming Tree?

For our PrintMaster application, most managed object classes that pertain to automobile makes will have the same naming structure. But if that were not the case, the naming structure for any given managed object class may be obtained from Metadata Services via the **getDNStruct** action. This action takes as the input argument the registration of the managed object class whose naming structure is to be retrieved. The output of the **getDNStruct** action is a list of the distinguishing attributes for that managed object class.

This information would then be used in completing the Managed Object Instance OM attribute of the CMIS_GET_ARGUMENT class.

16.7 Actions Supported by Metadata Services

The following is a list of the actions supported by Metadata Services:

getDNStruct retrieve naming structure for a given managed object class

getAttrStructList retrieve list of attributes supported by a given object class and information about the syntax for those attributes.

getSyntaxStruct retrieve information about the syntax for a given ASN.1 syntax definition

getRegIdsFromTemplNames retrieve the registration IDs given corresponding template names.

getTemplNamesFromRegIds retrieve the template names given corresponding registration ids

getPkgList retrieve the list of mandatory packages supported by a given managed object class.

getCondPkgList retrieve the list of conditional packages supported by a given managed object class.

getModuleNamesAndSyntax retrieve all module names currently stored in the Metadata Database and return a list of the ASN.1 syntax definition for each module.

getDocNamesAndObjs retrieve all document names currently stored in the Metadata Database and return a list of managed object classes for each document.

getAsn1EnumLabel retrieve field label corresponding to an ASN.1 value assignment of type ENUMERATED.

getAsn1OidLabel retrieve template name corresponding to an ASN.1 value assigment of type OBJECT IDENTIFIER.

getRegIdsAndLabels retrieve all the registration ids currently stored in the Metadata Database. For each registration id, return the qualified name of the corresponding template.

16.8 PrintMaster MIB File

This section contains a snapshot of the GDMO MIB file file for PrintMaster Version 1.0 Note that this is NOT the entire MIB but only a snippet.

```
.
--
-- PrintMaster Sample MIB
-- Version 1.0
--

--% DOCUMENT = "PrintMaster Version 1.0 : 1993"
--% REGISTERED AS {1 3 6 1 3 678 1}
--% NICKNAME = PX
--% STATUS =   New

   PX  OM-PACKAGE
    ABBREVIATION PX
    INITIAL ATTRIBUTE VALUE 20000
    ::= {1 3 6 1 3 678 99 }

manufacturerIdDesc   BEHAVIOR
   DEFINED AS  "The manufacturer's id. The id is a 2 digit number in the range
            of 1 to 99. The manufacturer id may be obtained by registering
            with the U.S. Department of Commerce." ;

manufacturerId   ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  PX-ASN1-Module.ManufacturerId ;
   MATCHES FOR          EQUALITY;
   BEHAVIOR            manufacturerIdDesc;
   REGISTERED AS          {1 3 6 1 3 678 99 7 1 } ;

modelIdDesc         BEHAVIOR
   DEFINED AS  "The model's id. The id is a 3 digit number in the range of
            of 1 to 999. The manufacturer is responsible for maintaining
            the list of model ids.";
modelId        ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  PX-ASN1-Module.ModelId ;
   MATCHES FOR          EQUALITY;
   BEHAVIOR            modelIdDesc;
   REGISTERED AS          {1 3 6 1 3 678 99 7 2 } ;

paperBinStatusDesc   BEHAVIOR
   DEFINED AS  "The status of the paper bin. Status is shown as a percentage
            remaining. ";

paperBinStatus   ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  PX-ASN1-Module.PaperBinStatus ;
   MATCHES FOR          EQUALITY;
   BEHAVIOR            paperBinStatusDesc;
   REGISTERED AS          {1 3 6 1 3 678 99 7 3 } ;

printerNameDesc      BEHAVIOR
   DEFINED AS  "The name of the printer. The name is assigned by the
            NetWork Administrator";

printerName      ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  PX-ASN1-Module.PrinterName ;
   MATCHES FOR          EQUALITY;
   BEHAVIOR            printerNameDesc;
   REGISTERED AS          {1 3 6 1 3 678 99 7 4 } ;

tonerStatusDesc     BEHAVIOR
   DEFINED AS  "The status of the printer's toner supply. Status is
            shown as a percentage remaining. ";

tonerStatus      ATTRIBUTE
   WITH ATTRIBUTE SYNTAX  PX-ASN1-Module.TonerStatus;
   MATCHES FOR          EQUALITY;
   BEHAVIOR            tonerStatusDesc;
   REGISTERED AS          {1 3 6 1 3 678 99 7 5 } ;
```

Figure 22 (Part 1 of 3). Snippet of PrintMaster Version 1.0 MIB File

```
resetPrinterDesc    BEHAVIOR
  DEFINED AS  "Reset the printer.";

resetPrinter    ACTION
  BEHAVIOR              resetPrinterDesc;
  REGISTERED AS         {1 3 6 1 3 678 99 9 1 } ;

stopPrinterDesc    BEHAVIOR
  DEFINED AS  "Stop the printer.";

stopPrinter    ACTION
  BEHAVIOR              stopPrinterDesc;
  REGISTERED AS         {1 3 6 1 3 678 99 9 2 } ;

printerPkgDesc              BEHAVIOR
  DEFINED AS  "This class models a typical printer. It provides the common
          attributes of a printer.";

printerPkg          PACKAGE
  BEHAVIOR          printerPkgDesc;
  ATTRIBUTES          modelId         GET,
                manufacturerId   GET,
                printerName      GET,
                paperBinStatus   GET,
                tonerStatus      GET;
   ACTIONS          resetPrinter,stopPrinter;
REGISTERED AS       {1 3 6 1 3 678 99 4 1 };

printer       MANAGED OBJECT CLASS
  DERIVED FROM
    "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
  CHARACTERIZED BY   printerPkg ;
REGISTERED AS      {1 3 6 1 3 678 99 3 1 };

laserPrinterPkgDesc        BEHAVIOR
  DEFINED AS  "This class models a standard laser printer. ";

laserPrinterPkg        PACKAGE
  BEHAVIOR          laserPrinterPkgDesc;
  ATTRIBUTES          paperBinStatus   GET,
                tonerStatus      GET;
REGISTERED AS       {1 3 6 1 3 678 99 4 2 };
```

Figure 22 (Part 2 of 3). Snippet of PrintMaster Version 1.0 MIB File

```
laserPrinter        MANAGED OBJECT CLASS
  DERIVED FROM
    "PrintMaster Version 1.0 : 1993":printer;
  CHARACTERIZED BY   laserPrinterPkg ;
REGISTERED AS      {1 3 6 1 3 678 99 3 2 };

laserPrinterNBDesc     BEHAVIOR
  DEFINED AS  "The distinguishing attribute for this managed object class
          is the printer's Name.";

laserPrinterNB   NAME BINDING
  SUBORDINATE OBJECT CLASS  laserPrinter;
  NAMED BY
  SUPERIOR OBJECT CLASS
    root;
  WITH ATTRIBUTE          printerName;
  BEHAVIOR              laserPrinterNBDesc;
  REGISTERED AS           {1 3 6 1 3 678 99 6 1 };

-- ASN.1 definitions referenced by the GDMO MIB itself.
--


PX-ASN1-Module DEFINITIONS ::= BEGIN


ManufacturerId ::= INTEGER ( 1..99 )

ModelId ::= INTEGER ( 1..999 )

PaperBinStatus ::= Percentage

Percentage ::= INTEGER ( 0 .. 100 )

PrinterName  ::= OCTET STRING ( SIZE (128) )

TonerStatus  ::=      Percentage

END
```

Figure 22 (Part 3 of 3). Snippet of PrintMaster Version 1.0 MIB File

16.9 Content Package Header File

This section contains a snapshot of the XMP content package header file for the PrintMaster Version 1.0 MIB file. Note that this is NOT the entire header file but only a snippet.

```
/*********************************************************************
 * MODULE NAME= px.h
 * FUNCTION= Header file for IBM LAN NetView XMP Contents Package
\***********************-END OF SPECIFICATIONS-************************/

/* PX Package Definition */
#define OMP_O_PX        "\x2B\x06\x01\x03\x85\x26\x63"

/* Intermediate Object Identifier macro */
#define mpP_px(X)       (OMP_O_PX#X)

/* PX Managed Object OID Definitions */
#define OMP_O_PX_O_LASER_PRINTER        "\x2B\x06\x01\x03\x85\x26\x63\x03\x02"
#define OMP_O_PX_O_PRINTER      "\x2B\x06\x01\x03\x85\x26\x63\x03\x01"

/* PX Attribute OID Definitions */
#define OMP_O_PX_A_MANUFACTURER_ID      "\x2B\x06\x01\x03\x85\x26\x63\x07\x01"
#define OMP_O_PX_A_MODEL_ID     "\x2B\x06\x01\x03\x85\x26\x63\x07\x02"
#define OMP_O_PX_A_PAPER_BIN_STATUS     "\x2B\x06\x01\x03\x85\x26\x63\x07\x03"
#define OMP_O_PX_A_PRINTER_NAME "\x2B\x06\x01\x03\x85\x26\x63\x07\x04"
#define OMP_O_PX_A_TONER_STATUS "\x2B\x06\x01\x03\x85\x26\x63\x07\x05"

/* PX Attribute Group OID Definitions */

/* PX Notification OID Definitions */

/* PX Action OID Definitions */
#define OMP_O_PX_T_RESET_PRINTER        "\x2B\x06\x01\x03\x85\x26\x63\x09\x01"
#define OMP_O_PX_T_STOP_PRINTER "\x2B\x06\x01\x03\x85\x26\x63\x09\x02"

/* PX Parameter OID Definitions */

/* PX Name Binding OID Definitions */
#define OMP_O_PX_B_LASER_PRINTER_NB     "\x2B\x06\x01\x03\x85\x26\x63\x06\x01"

/* PX Package OID Definitions */
#define OMP_O_PX_P_LASER_PRINTER_PKG    "\x2B\x06\x01\x03\x85\x26\x63\x04\x02"
#define OMP_O_PX_P_PRINTER_PKG  "\x2B\x06\x01\x03\x85\x26\x63\x04\x01"

/* PX Relationship Binding OID Definitions */

/* PX Relationship Class OID Definitions */

/* OM Class Object Identifiers */


/* Attribute Name Constants */


/* Value Lists */
```

Figure 23. Snippet of XMP Content Package Header File for PrintMaster Version 1.0

16.10 Content Package Source File

This section contains a snapshot of the XMP content package source file for the PrintMaster Version 1.0 MIB file. Note that this is NOT the entire source file but only a snippet.

```
/*************************************************************************
 * MODULE NAME= px.c
 * FUNCTION= Source file for IBM LAN NetView XMP Contents Package
\***********************-END OF SPECIFICATIONS-*************************/

#include <xom.h>
#include <pkg_hdr.h>

#include <px.h>

#define CONV_TO_OM_STRING(a) { (sizeof(a) - 1), (a) }


/* START OF PACKAGE PX */

/* START OF PACKAGE ATTRIBUTE ENTRY 0 */

/* PACKAGE ATTRIBUTE _PX_package_attribute_7 */

static IMP_ATTRIBUTE _PX_package_attribute_7 = {
   (OM_type) 0,              /* attr_type */
   OM_S_INTEGER,             /* attr_syntax */
   "Manufacturer-Id",        /* pname */
   {IMP_NO_DEFINED_LIMITS,           /* min */
   IMP_NO_DEFINED_LIMITS},           /* max */          /* limits */
   IMP_NO_DEFINED_LIMITS,            /* mva_max */
   NULL,           /* attr_copy */
   NULL,           /* attr_destroy */
   NULL,           /* attr_get */
   NULL,           /* attr_put */
   NULL,           /* attr_verify */
   IMP_EXPAND_EXPLICIT,              /* expand */
   IMP_TAG_NONE,             /* tag_type */
   0x0        /* tag_value */
};

/* PACKAGE ATTRIBUTE ENTRY _PX_package_attribute_7 */

static IMP_PACKAGE_ATTR_ENTRY _PX_package_attr_entry_6 = {
   CONV_TO_OM_STRING(OMP_O_PX_A_MANUFACTURER_ID),           /* oid */
   &_PX_package_attribute_7          /* pdata */
};

/* END OF PACKAGE ATTRIBUTE ENTRY 0 */

/* START OF PACKAGE ATTRIBUTE ENTRY 1 */

/* PACKAGE ATTRIBUTE _PX_package_attribute_9 */

static IMP_ATTRIBUTE _PX_package_attribute_9 = {
   (OM_type) 0,              /* attr_type */
   OM_S_INTEGER,             /* attr_syntax */
   "Model-Id",         /* pname */
   {IMP_NO_DEFINED_LIMITS,           /* min */
   IMP_NO_DEFINED_LIMITS},           /* max */          /* limits */
   IMP_NO_DEFINED_LIMITS,            /* mva_max */
   NULL,           /* attr_copy */
   NULL,           /* attr_destroy */
   NULL,           /* attr_get */
   NULL,           /* attr_put */
   NULL,           /* attr_verify */
   IMP_EXPAND_EXPLICIT,              /* expand */
   IMP_TAG_NONE,             /* tag_type */
   0x0        /* tag_value */
};

/* PACKAGE ATTRIBUTE ENTRY _PX_package_attribute_9 */

static IMP_PACKAGE_ATTR_ENTRY _PX_package_attr_entry_8 = {
   CONV_TO_OM_STRING(OMP_O_PX_A_MODEL_ID),          /* oid */
   &_PX_package_attribute_9          /* pdata */
};

/* END OF PACKAGE ATTRIBUTE ENTRY 1 */
```

Figure 24 (Part 1 of 3). Snippet of XMP Content Package Source File for PrintMaster Version 1.0

```
/* START OF PACKAGE ATTRIBUTE ENTRY 2 */

/* PACKAGE ATTRIBUTE _PX_package_attribute_11 */

static IMP_ATTRIBUTE _PX_package_attribute_11 = {
    (OM_type) 0,              /* attr_type */
    OM_S_INTEGER,             /* attr_syntax */
    "Paper-Bin-Status",       /* pname */
    {IMP_NO_DEFINED_LIMITS,          /* min */
    IMP_NO_DEFINED_LIMITS},          /* max */          /* limits */
    IMP_NO_DEFINED_LIMITS,           /* mva_max */
    NULL,             /* attr_copy */
    NULL,             /* attr_destroy */
    NULL,             /* attr_get */
    NULL,             /* attr_put */
    NULL,             /* attr_verify */
    IMP_EXPAND_EXPLICIT,             /* expand */
    IMP_TAG_NONE,             /* tag_type */
    0x0       /* tag_value */
};
/* PACKAGE ATTRIBUTE ENTRY _PX_package_attribute_11 */

static IMP_PACKAGE_ATTR_ENTRY _PX_package_attr_entry_10 = {
    CONV_TO_OM_STRING(OMP_O_PX_A_PAPER_BIN_STATUS),          /* oid */
    &_PX_package_attribute_11          /* pdata */
};

/* END OF PACKAGE ATTRIBUTE ENTRY 2 */


/* START OF PACKAGE ATTRIBUTE ENTRY 3 */

/* PACKAGE ATTRIBUTE _PX_package_attribute_13 */

static IMP_ATTRIBUTE _PX_package_attribute_13 = {
    (OM_type) 0,              /* attr_type */
    OM_S_OCTET_STRING,        /* attr_syntax */
    "Printer-Name",           /* pname */
    {IMP_NO_DEFINED_LIMITS,          /* min */
    IMP_NO_DEFINED_LIMITS},          /* max */          /* limits */
    IMP_NO_DEFINED_LIMITS,           /* mva_max */
    NULL,             /* attr_copy */
    NULL,             /* attr_destroy */
    NULL,             /* attr_get */
    NULL,             /* attr_put */
    NULL,             /* attr_verify */
    IMP_EXPAND_EXPLICIT,             /* expand */
    IMP_TAG_NONE,             /* tag_type */
    0x0       /* tag_value */
};

/* PACKAGE ATTRIBUTE ENTRY _PX_package_attribute_13 */

static IMP_PACKAGE_ATTR_ENTRY _PX_package_attr_entry_12 = {
    CONV_TO_OM_STRING(OMP_O_PX_A_PRINTER_NAME),       /* oid */
    &_PX_package_attribute_13          /* pdata */
};

/* END OF PACKAGE ATTRIBUTE ENTRY 3 */

/* START OF PACKAGE ATTRIBUTE ENTRY 4 */

/* PACKAGE ATTRIBUTE _PX_package_attribute_15 */

static IMP_ATTRIBUTE _PX_package_attribute_15 = {
    (OM_type) 0,              /* attr_type */
    OM_S_INTEGER,             /* attr_syntax */
    "Toner-Status",           /* pname */
    {IMP_NO_DEFINED_LIMITS,          /* min */
    IMP_NO_DEFINED_LIMITS},          /* max */          /* limits */
    IMP_NO_DEFINED_LIMITS,           /* mva_max */
    NULL,             /* attr_copy */
    NULL,             /* attr_destroy */
    NULL,             /* attr_get */
    NULL,             /* attr_put */
    NULL,             /* attr_verify */
    IMP_EXPAND_EXPLICIT,             /* expand */
    IMP_TAG_NONE,             /* tag_type */
    0x0       /* tag_value */
};
```

Figure 24 (Part 2 of 3). Snippet of XMP Content Package Source File for PrintMaster Version 1.0

```
/* PACKAGE ATTRIBUTE ENTRY _PX_package_attribute_15 */

static IMP_PACKAGE_ATTR_ENTRY _PX_package_attr_entry_14 = {
   CONV_TO_OM_STRING(OMP_O_PX_A_TONER_STATUS),        /* oid */
   &_PX_package_attribute_15          /* pdata */
};

/* END OF PACKAGE ATTRIBUTE ENTRY 4 */


/* PACKAGE ATTRIBUTE TABLE PX */

static IMP_ATTR_ENTRY_ADDR _PX_package_attr_table_2 [] = {
&_PX_package_attr_entry_6,
&_PX_package_attr_entry_8,
&_PX_package_attr_entry_10,
&_PX_package_attr_entry_12,
&_PX_package_attr_entry_14,
};

/* PACKAGE PX */

IMP_PACKAGE_ENTRY PX_pkg_table = {
   {NULL,NULL,NULL},          /* ll */
   IMP_PACKAGE_VERSION,              /* status */
   "PX",             /* pname */
   CONV_TO_OM_STRING(OMP_O_PX),              /* oid */
   0x0,           /* min_class_subid */
   0x0,           /* num_class */
   NULL,            /* pclass_tbl */
   0x0,           /* min_attr_subid */
   0x5,           /* num_attr */
   &_PX_package_attr_table_2[0],           /* pattr_tbl */
   0x0,           /* min_action_subid */
   0x0,           /* num_action */
   NULL,            /* paction_tbl */
   0x0,           /* min_event_subid */
   0x0,           /* num_event */
   NULL,            /* pevent_tbl */
   0x0,           /* min_param_subid */
   0x0,           /* num_param */
   NULL             /* pparam_tbl */
};

/* END OF PACKAGE PX */
```

Figure 24 (Part 3 of 3). Snippet of XMP Content Package Source File for PrintMaster Version 1.0

16.11 Summary

This paper has shown the two basic types of managed applications

- specialized management applications
- generic management applications

It has shown how specialized management applications, (that is, PrintMaster Version 1.0) make use of the content package deliverables via the OM_EXPORT macro to setup the various OM data structures passed on input to XMP function calls. It has also shown how the same content package deliverables are used in parsing through the results from a particular mp_get_req or mp_action_req.

Generic management applications (that is, PrintMaster Version 2.0) offer an entirely different paradigm. Here the management application can handle a dynamic list of managed object classes whose definitions are allowed to change. The application when designed properly makes use of Metadata Services and utilizes several XMP convenience routines to obtain the information which is no longer readily available.

It was pointed out that the complexity of a generic managed application is quite challenging but the resultant possibilities are equally impressive.

Part 4. ISV-Related Products and Information

Chapter 17. Opportunities for LAN NetView-based Applications

This paper explains the decision points for an ISV or service provider to consider in choosing to build LAN NetView-based offerings. It identifies the overall opportunity/market and points out how an ISV or service provider could profit by such an investment. It also describe s assistance that IBM can provide to help the ISV or service provider to be successful (developer's conferences, early code, complementary efforts, marketing, etc.).

17.1 Scope

This article is aimed at application developers, with the intent of justifying the IBM LAN NetView platform as a framework for development and implementation of systems and network management applications. It answers the basic question "Why invest in LAN NetView offerings?".

17.2 Introduction

Today LAN-based systems are providing an increasing number of business solutions, and the need for effective management of the hardware and software resources that comprise these systems; keeping them operational and controlling administrative costs, has become more important than ever. Couple this with the need to reduce people costs, while at the same time, trying to provide a responsive information system, and you can see that management has a strong challenge to meet.

Coincidentally there has been a a greater focus on LAN (and WAN) management over the past few years. However, this focus has been primarily in the area of the communications components: adapters, wire, hubs, bridges, routers, etc., with little attention being paid to managing the workstations and the software operating on them. This has become a problem in today's networking environment where software errors can cause severe bottlenecks and result in system and network downtimes.

17.3 Defining the Problem: The Downside of Downsizing

The shift from a centralized mainframe environment to the distributed personal systems environment has resulted in a number of benefits related to improvement of user productivity and cost savings, such as more efficient development of applications, local autonomy, and sharing of resources. However, as these benefits were being realized, the benefits of managing system resources from a central point was lessened. Providing Asset Control and Security, Backup and Recovery, Problem Management, Performance Management, et al, has become more difficult because of the remoteness of the systems that need to be managed. Whereas in the past an I/S organization provided these functions in an end-to-end systems management paradigm, in many cases these functions are now being left to the end-user, or remote LAN administrator.

With this in mind, it's apparent that the need exists for a comprehensive systems management solution to the problem of managing I/S resources in a distributed environment. A distributed systems management approach to monitor and control the I/S components helps meet the challenge of managing both the hardware and software resources by providing functions for the management disciplines mentioned above; such as Asset Management, Backup/Restore, Change Control, Performance Monitoring, and Problem Determination - IN A CLIENT/SERVER ENVIRONMENT.

The next logical question would be: what solutions are out there today? While there are several systems and network management applications available today which implement these management disciplines, many provide partial solutions; data often cannot be shared, the end-user interfaces are inconsistent in their presentation, management services rely on proprietary connections, and in general, the integration of such applications is left to the customer.

17.4 Defining the Solution: An O P E N Approach

How does IBM LAN NetView product address these shortcomings? By providing an "open" approach to distributed systems management. The major factors that contribute to this approach are:

1. Comprehensive management services, in terms of topology/discovery, object management, communications services, event management, and metadata services. The management services are based on The HP OpenView Network Management Server 3.1. By providing the underlying management services, application developers can concentrate on application function rather than the access mechanism for managing system resources.
2. An industry standard "open" (API) that provides transparent access to these underlying management services. The API used for accessing the services is the X/Open Management Protocol, which was selected by the OSF for the Distributed Management Environment (DME). The openness afforded by using an industry standard API adds value to the application, providing a measure of portability to other operating system platforms. The portability can be extended within the range of IBM products to include the AIX SystemView NetView/6000* platform, which also implements the XMP API.

The LAN NetView product also provides a common graphical end user interface in the LAN NetView Manage product that is referred to as View. View uses an object-oriented approach to managing graphical objects; using OS/2 and implementing System Object Model (SOM), an object-oriented library technology. One of the advantages of this SOM-based approach from an application development perspective is that it provides a seamless integration of the management applications. This results in consistency for both the application user, and the management application.

17.5 Why OS/2?

To multitask or not multitask, that is the question. Whether or not a multitasking operating system is really needed in cases where the end users are typically "single tasking" workers has been questioned in the past. However, as a true pre-emptive multitasking operating system, OS/2 has been well received as an operating system for "critical" applications. What's more critical than controlling the operation and assets on your Information System network? Systems management can be more effectively provided through the use of the multitasking capabilities inherent in OS/2. multitasking capabilities. Do you want to report a problem automatically to a managing system while the workstation continues to perform normal activities? Do you want to apply changes to the software on the workstation while other workstation operations continue? Do you want to analyze the performance on the workstation, its DASD, memory, storage, etc., while processing critical transactions? A multitasking operating system is best suited for non-intrusively performing such systems and network management tasks.

Development benefits of using OS/2 can also be realized. The OS/2 Developer's Assistance Program offers a broad range of services including:

- Technical assistance through OS/2 fora on the CompuServe information service
- Access to early code
- OS/2 application migration workshops and seminars
- Online technical support
- OS/2 marketing programs
- Consideration for participation in trade shows
- Access to OS/2 development tools
- Hardware/Software rebates and loaner equipment
- IBM Direct Marketing Center

It is IBM's objective to make OS/2 the best managed workstation platform in the industry. The LAN NetView product, through it's open-architected platform, and CMIP agents for the OS/2 operating system, and LAN Server 3.0, LAN Requester, Database Manager, and Communications Manager products, is the vehicle for achieving this objective.

17.6 LAN NetView Applications: The Opportunity

17.6.1 Integration

I believe it was George Washington who said "Though the first step in providing a solution for distributed systems management is delivering a solid platform, it's the applications running on that platform that provide the total solution." Maybe it wasn't George Washington, but it's still true. LAN NetView applications from IBM and ISVs will deliver this solution by integrating network management, performance analysis, problem management, configuration management, asset management, and related functions, with the LAN NetView platform.

17.6.2 Cost Containment

At a time where more and more enterprises are downsizing, or considering downsizing, the cost of managing their LANs and WANs is a major factor. The LAN NetView product provides a comprehensive management platform, using an OS/2 2.x 32-bit, multitasking operating system, at a cost which is significantly less than a UNIX-based platform.

17.6.3 Workstation Management

When you consider that SNMP is supported on the platform to accommodate a wide variety of device management, and add to that the additional capability which the LAN NetView CMIP agents offer; that is the ability to manage the workstations themselves, a significant advantage is provided to the applications that use the LAN NetView platform.

17.6.4 DOS and Windows Support

DOS systems and DOS systems using the Microsoft Windows product also need to be managed. The LAN NetView family of products includes agents to manage IBM DOS 5.0 and 6.1, Microsoft DOS 5.0 and 6.0, and DOS + Windows 3.0 and 3.1. These agents can be utilized by management applications on a managing system, using CMOL to request data about the resource managers, and manage the resource managers by setting and changing values.

17.6.5 Host Gateway Support

Is it important for your application or agent to communicate to NetView on the host? This provision is accommodated through the LAN NetView Tie application, which acts as a gateway to/from NetView. Thus, in addition to managing interconnected workgroup LANs, LAN NetView applications can prosper in a large enterprise where a Manager of Managers system is the controlling point.

17.7 Decisions, Decisions, Decisions

For application developers, the choice of a systems/network management platform needs to be based of course on Return-On-Investment. Let's look at some of the decision points an ISV would need to consider in choosing the right platform to support:

- WHAT MARKET DOES THE PLATFORM ADDRESS?

  Will applications on this platform sell in the workgroup LAN environment as well as in a large enterprise?

  The LAN NetView product is targeted at the LAN workgroup segment, yet plays a significant role in large enterprises where NetView on a mainframe can manage LAN NetView-attached resources via the LAN NetView Tie application.

- DOES THE PLATFORM AFFORD PORTABILITY TO OTHER PLATFORMS?

  Is an open architecture implemented, using industry standards?

  Using an "open system" approach was key to the LAN NetView framework development. The LAN NetView platform uses an open API (XMP; endorsed by the Open Software Foundation (OSF) and by the IBM SystemView strategy), components from the OSF DME (Distributed Management Environment), standard management and transport protocols (CMIP, SNMP, TCP/IP and 802.2), and GDMO (ISO Guidelines for the Definition of Managed Objects). This facilitates easier porting to other platforms using these standards.

- DOES IT PROVIDE THE BASIS FOR TOTAL SYSTEMS MANAGEMENT?

Does it provide the capability to manage workstations as well as network connections?

One of the major salient points of the LAN NetView product is its ability to manage workstations with either OS/2 2.x, DOS, or DOS with the Microsoft Windows product installed, using agents that are included in the LAN NetView family of products. In addition, the LAN NetView product also supports SNMP, which is used primarily for network device management.

- IS IT EASY TO WORK WITH?

  Does the development environment use object-oriented technology, which generally involves less coding? Is transparency provided to the developer for access to the management services? Are development tools provided? Are the APIs standard ones?

  The View component of LAN NetView Manage version 1.0 uses a SOM (System Object Model)-based GUI (Graphical User Interface) which provides an object-oriented mechanism for applications to interoperate with objects and other applications. This, along with an open API and the facilities of OS/2 2.x with its flat memory model and powerful 32-bit, multitasking capabilities provide an excellent development environment.

- IS IT A STRATEGIC OFFERING?

  Is it part of a reliable company's overall systems management strategy, or is it a tactical offering?

  SystemView is the structure that represents IBM's systems management strategy, and has been well-received by customers as a comprehensive model for implementing systems management solutions across all IBM platforms. The LAN NetView product is the OS/2 implementation of IBM's SystemView strategy, and is compliant across all the SystemView dimensions; End-Use, Data, and Application. The LAN NetView product is IBM's strategic systems management platform on the personal workstation.

- IS IT COST EFFECTIVE?

  How does it compare to other platforms in terms of price/performance?

  When compared to UNIX-based systems management offerings, the LAN NetView family of products offers a lower cost alternative to managing systems and network resources, from a lower cost hardware base.

- WHAT TYPE OF ISV SUPPORT IS OFFERED?

  Is there a support program in place that can help with problems and marketing support, or is it simply "here's the code and documentation"?

  Since LAN NetView is an OS/2-based platform, developers can take advantage of the many OS/2 marketing and technical assistance offerings through the IBM OS/2 Worldwide Developers Assistance Program.

- WHAT APPLICATIONS ARE ON THE PLATFORM?

  Are the base applications competitive? Will my application complement them?

  Included in the IBM LAN NetView family of products are: LAN NetView Monitor, a comprehensive performance monitor for systems with OS/2; LAN NetView Fix, a problem management application capable of handling both OSI alarms and SNMP Trap conditions, with various alarm/event notification functions; LAN NetView Scan, an application for asset management of workstation vital product data; and LAN NetView Tie, a gateway to NetView

on the mainframe, converting alarms to NetView alerts and forwarding them on to NetView for further processing.

- IS THIS A PLATFORM FOR GROWTH?

  Is there a commitment to provide enhancements in terms of additional management capabilities?

  Again, the LAN NetView product is IBM's strategic systems management platform on OS/2. Therefore it is expected that it will be enhanced based on improvements in technology and standards that are developed within the systems and network management arena.

- IS THERE A STRONG MARKETING FORCE BEHIND THIS PLATFORM?

  Are all marketing channels fully exploited for this platform?

  The LAN NetView family of products comprise a solution for LAN workgroups in all types of enterprises; functioning as a self-contained management system in small enterprises, and as management segments interoperating with a Manager-of-Managers system such as NetView on the host in a large enterprise environment. This broadens the marketing opportunity to all segments where LAN-based systems are used.

This of course is not an all-inclusive list; other considerations include the skill levels of the developers, resource constraints, specific application target areas, marketing objectives, training and education requirements, et al. The point is that there are a number of factors to ponder when choosing where to make the investment, and how much of an investment to make in developing or porting an application to a platform.

17.8 LAN NetView and ISV Applications: The Right Stuff

IBM's leadership role in network management can clearly be seen through the success of both the NetView and NetView/6000 products. The LAN NetView product builds on these successful management platforms providing distributed systems management on the personal workstation. Now, in addition to providing centralized and distributed management of network components from Host and RISC-based systems, LAN Administrators can manage the personal workstations themselves, from a lower-cost, Intel-based, LAN NetView product.

The success of the LAN NetView platform can only be achieved with the support of ISVs through their systems and network management applications, integrated with the LAN NetView platform. There's an interdependence here; the LAN NetView platform providing the basis for managing the system resources, and the applications carrying out the management functions.

As a LAN NetView application provider, you are helping to extend the platform function, and in turn, your application can broaden its market reach through IBM promotions, such as those offered through the IBM OS/2 Developer Assistance Program.

In addition, remember this is not simply a development platform; LAN NetView Manage and Enabler provide significant management capabilities through the management services, APIs, user-interface, topology/discovery, and agent functionality. Together with applications from IBM and ISVs, this functionality will enable customers to enhance availability of their network, increase their productivity, and accomplish this with a cost-effective solution.

17.9 LAN NetView ISV Support Program: Need Help?

In order to encourage ISVs to port existing applications and/or develop new applications on the LAN NetView platform, IBM provides assistance in the form of:

- Early code and documentation
- Tutorials and workshops
- Online support through the CompuServe information service and IBM fora
- Toll-Free Hotline support for technical assistance
- OS/2 Marketing Program support via Developer Assistance Program

Note: This is in addition to the support provided through the OS/2 Developer's Assistance Program mentioned earlier.

17.10 Summary: The Points of Light

Let's look at some of the salient points of the LAN NetView product in the context of providing application developers with the means for contributing to the total solution referred to above:

- THE MARKET

  The LAN NetView family of products focus on the workgroup LANs and interconnected LANs. However, because of the capabilities offered that allow control of LAN NetView systems by a NetView Console Operator on the Host, there is a significant area of opportunity for LAN NetView applications in the large enterprise environments.

  The LAN NetView product offers a low-cost solution to managing LAN-based systems.

- PORTABILITY

  The LAN NetView platform uses selected portions of the DME technology, and the XMP API for access to the underlying management services, and SOM. This facilitates an easier porting effort from the LAN NetView platform to systems such as AIX SystemView NetView/6000 product from IBM, and the OpenView system from Hewlett-Packard.

- STRATEGIC OFFERING

  The LAN NetView product is part of the NetView family of products from IBM, and is the OS/2 implementation of major components of IBM SAA SystemView, IBM's strategy for enterprise-wide systems management.

- DEVELOPMENT/RUNTIME ENVIRONMENT

  OS/2, with it's flat memory model and powerful 32-bit multitasking capabilities provides an excellent development and runtime environment. This, along with the transparency to either CMIP or SNMP objects provided through the XMP API, and an object-oriented user-interface (View) provide tools that aid developers.

- ISV SUPPORT

  Access to early code and documentation, as well as education and marketing programs are provided, with toll-free technical help available from the development organization.

In review, we've looked at some of the problems associated with maintaining the operation and controlling the costs of LAN-based Information Systems. This problem is heightened with the trend toward downsizing and distributed processing on personal systems. In general, the opportunity exists for a comprehensive systems management platform and applications which, together can manage both the hardware and software components comprising the Information System.

Notice once again the reference to systems management rather than just network management. The importance of this, particularly in a distributed environment can't be over emphasized. When you consider the value of the software assets in an enterprise, and the need for accountability for these assets, as well as functions such as capacity planning, configuration management, and problem management, you can see how a system that only manages the network devices and connectivity falls short of providing a total solution.

Chapter 18. LAN NetView Catalog of Value-added Offerings

This document is intended to be used as a catalog of offerings that add value to the core LAN NetView systems management platform and functions. It is intended for IBM marketing representatives and support personnel, consultants, trade press, customers and third party developers. The purpose of this document is to provide information regarding the complete range of capabilities available (or soon to be available) with the LAN NetView family of products.

18.1 Introduction

This group of product offerings for the LAN NetView platform is the initial list of available and committed offerings at the time of the LAN NetView family of products availability (October 1993).

An overview of the LAN NetView family of products is provided, immediately following the introduction, which briefly describes the functions included in the LAN NetView systems management platform.

The 18.3, "LAN NetView Product and Service Offerings" on page 265 describes the products and services available, or soon to be available, from Independent Software Vendors (ISVs), and from IBM.

A section titled "Complimentary Products from IBM" lists those products that provide value-added function to the LAN NetView family of products which are not integrated with the LAN NetView platform, but do provide needed systems and network management function by coexisting with the LAN NetView products.

Additional information about vendor products can be obtained by contacting the appropriate vendor location identified in this document.

18.2 LAN NetView Overview

The IBM LAN NetView family of products provides a standards-based platform for the creation and implementation of systems management applications for the LAN workgroup environment. The management framework for the products is an OS/2 implementation of technology selected by the Open Software Foundation (OSF) for the Distributed Management Environment (DME). The platform services and applications allow the management of LAN-attached clients and servers running the OS/2 2.x, IBM DOS 5.0 and 6.1, Microsoft DOS 5.0 and 6.0, and DOS with Microsoft Windows 3.0 or 3.1 operating systems.

Within the IBM NetView family of products (NetView, NetView/6000, and LAN NetView), LAN NetView is positioned as the systems management platform of choice for the small to medium size enterprises where the customer's installed base is primarily personal workstations (for example, PS/2s and compatibles), and the number of network nodes to be managed is less than 500, per managing system.

The LAN NetView platform architecture is based on the OSI (Open Systems Interconnection) Systems Management Model, which defines:

1. A managing system with the following attributes:

- Has a user interface.
- Executes the management applications.
- Performs a standard set of actions on managed objects.

2. A managed system on which an agent or agents monitor and control managed resources and provide notifications of events to the managing system.

The managing system in a LAN NetView configuration would have the LAN NetView Manage version 1.0 product and agents installed on it. Any managed OS/2 systems would have LAN NetView Enabler version 1.0 and agents installed, and DOS systems (as well as DOS with Microsoft Windows 3.0 or 3.1) would have LAN NetView Agents for DOS installed. Remote management of the LAN resources is enabled by LAN NetView's ability to accept commands from a command line or application program running on the Managing system, from NetView via SNA connection, from another OS/2 system with the Distributed Console Access Facility (DCAF) installed, and/or from a remote system asynchronously connected to the managing system using the IBM LAN Distance product.

There's a substantial value-add here that the LAN NetView product brings to LAN management; it's a platform for systems management and network management. Network management is generally thought to refer to the management of SNMP devices, such as bridges, routers, hubs, etc. The LAN NetView platform can be used for managing these devices as well, since the platform supports SNMP as well as CMIP. The Request Manager applet (mini-application), which is included in the LAN NetView Manage 1.0 product, can be used for this purpose.

18.3 LAN NetView Product and Service Offerings

18.3.1 Allen Systems Group, Inc.

18.3.1.1 IMPACT**

IMPACT is an open architecture data center automation tool that provides realtime management support by automating change management, inventory control, critical problem reporting and notification. In accordance with Allen Systems Group's commitment to IBM's SystemView systems management strategy, a client server version of IMPACT written to the standardized interfaces of the IBM LAN NetView platform is currently being developed.

```
Allen Systems Group
750 11th Street S.
Naples, Fl. 33940
Phone:  813/263-6700
or      800/93-ALLEN

Contact: Randy Cook
```

18.3.2 Allerion**, Inc.

18.3.2.1 SUPPORTlink Network Management Service

Allerion Inc. is a network integrator specializing in products and services for the management of networked technology, which encompasses both logical and physical network devices. Network management is an issue many companies are addressing today. In response, Allerion Inc. has developed a unique service offering called SUPPORTlink. Based on an integrated set of management tools, including LAN NetView and NetView/6000, SUPPORTlink provides a proactive network management service to detect network events, aid in their resolution and analyze performance. ,* SUPPORTlink is offered in two ways:

1. SUPPORTlink On-Site - Allerion will integrate the SUPPORTlink Network Control Center and the associated management tools on your site, train personnel in its operation and provide on-going support.
2. SUPPORTlink Remote - a service bureau approach that provides network support without investing resources in network management tools.

```
Allerion Inc.
717 Ridgedale Avenue
East Hanover, New Jersey 07936
Phone:  201/887-1000
FAX:    201/887-9546

Contact: Arnie McKinnis
```

18.3.3 BGS Systems, Inc.

18.3.3.1 Performance and Capacity Management for OS/2

- **BEST/1-Visualizer for OS/2****
- **Analyze for OS/2****

BEST/1-Visualizer for OS/2 and Analyze for OS/2 together provide centralized performance and capacity management capabilities for OS/2 servers and workstations. Analyze for OS/2 processes data from either LAN NetView Enabler or SPM/2 and presents it to BEST/1-Visualizer for OS/2 for storing, analyzing and communicating performance and capacity information. Both products are scheduled to be available from BGS in early 1994.

Analyze for OS/2 collects CPU, memory and disk resource usage by process and correlates LAN data to reveal the consumption of server resources by clients. Analyze for OS/2 supports the creation of application or business-oriented workloads to provide a much more meaningful perspective than is available from raw process-level data. Analyze for OS/2 is designed to manage multiple OS/2 nodes from a central point and can be easily configured for unattended automated daily operation. Users have complete control over which OS/2 nodes are managed and which performance metrics are collected.

BEST/1-Visualizer for OS/2 is a PC-based product consisting of a database of key performance and capacity information, extensive facilities for point-and-click hierarchical-zoom analysis, and automated customizable graphical and tabular reporting. Using BEST/1-Visualizer, organizations can manage their distributed OS/2 environment, as well as their other platforms, from a central PC.

BEST/1-Visualizer is a keystone in the BGS strategy to provide common tools for performance and capacity management across the diverse computing platforms used in today's enterprises. Versions of BEST/1-Visualizer are now generally available for MVS, VM and OpenVMS**. Future releases will support UNIX, Windows NT** and AS/400*.

BEST/1-Visualizer for OS/2 and Analyze for OS/2 fit into the BGS framework called Performance Assurance**. Performance Assurance is an enterprise-wide process which provides for:

- Solving Day-To-Day Problems
  - Generate exception reports automatically
  - Utilize graphical bottleneck analysis
- Tracking Long-Term Performance
  - Trend and forecast workload activity
  - Compare actual to planned performance
- Predicting The Impact of Change
  - Test drive tuning alternatives
  - Size client/server configurations
- Communicating With Management and Users
  - Provide support for recommendations
  - Just expenditures for resources

BGS Systems, Inc.
128 Technology Center
Waltham, MA 02254-9111
(617)891-0000
(617)890-0000 fax

Contact: Rose Mary Gabler

18.3.4 C.O.L. Systems, Inc.

18.3.4.1 Osrm2** 2.0

Osrm2 2.0 is a performance and capacity management system for OS/2 2.x systems that provides a centralized data repository of performance and capacity statistics captured on other machines running OS/2 1.x, OS/2 2.x, NetWare, and Windows NT. Through end-of-day processing and post-processing tools, the captured statistics can be analyzed for possible performance tuning and capacity management opportunities. Data can be exported through user hooks and IBM REXX/2 procedures into IBM DB2/2, IBM NetView, Sun Systems' SunNet Manager**, and IBM LAN NetView.

```
C.O.L. Systems, Inc.
Mill Pond Offices
Suite 105
Route 100
Somers, N.Y.  10589
Phone: (914) 277-4312

Contact: Frank Castelluci
```

18.3.5 Computer Associates International, Inc.

18.3.5.1 CA-UNICENTER

CA-UNICENTER is an integrated client-server systems management solution delivering substantial management capabilities in five critical areas including security, help desk, and problem management, file backup and archive, workload scheduling and console management.

With CA-UNICENTER's client-server architecture, organizations can distribute OS/2 Workplace Shell administrative clients throughout the network based on job function and geographic requirements while maintaining central control of all management policies through CA-UNICENTER's distributed database technology. CA-UNICENTER management benefits extend to user-written programs, third party applications, and systems management platforms like IBM LAN NetView through programmable interfaces. With CA-UNICENTER many systems management tasks can be automated according to enterprise management policies, reducing reliance on manual operations.

Availability: BETA/October, 1993 GA/December, 1993

```
Computer Associates International
1 Computer Associates Plaza
Islandia, New York 11788-7011
(516) 324-5224 x2391

Contact: Bob Gordon
```

18.3.6 Dolphin Networks

18.3.6.1 DOLPHIN ESP** Family of Protocol Analyzers:

- **Dolphin ESP Ethernet**
- **Dolphin ESP Token Ring**

Dolphin ESP is the powerful network protocol analyzer that provides a clear "window" for viewing Ethernet and Token-Ring problems. Each product incorporates technology to troubleshoot problems on a specific network topology: Ethernet or Token-Ring (4 and 16 MB/sec), with both offering powerful functionality and a very easy-to-use interface.

Each Dolphin ESP product consists of two elements: DOS-based software and a custom, 16-bit ISA network interface card compatible with the type of network being monitored. This hardware/software combination delivers the necessary high-performance needed by administrators to view and troubleshoot intensely-operated networks.

Major product features include:

Real-Time Performance - Critical to preventing costly problems on the network. All monitored data can be simultaneously stored to a hard disk for exhaustive post-analysis.

Real-Time Displays of Data: Maximum viewing flexibility in real-time. Quickly view and change among color-coded displays showing Bandwidth, Statistical, Utilization or Pairs data.

Extensive Protocol Support: Decodes all 7-layers of popular network operating system protocols. Supports over 100 protocols that include Novell's IPX/SPX, TCP/IP, Appletalk, BANYAN VINES**, DECnet**, IBM's NetBIOS, XNS**, SMB (Windows** NT**) and more.

```
Dolphin Networks
4405 International Blvd.,  Suite B-108
Norcross, Georgia 30093
(404)279-7050
(404)279-1615 fax

Contact: Phillip Kim
```

18.3.7 Gradient Technologies, Inc.

18.3.7.1 iFOR/LS**

Gradient's iFOR/LS is a systems management application license manager which can be administered using the LAN NetView framework. With iFOR/LS software vendors are assured they are fairly compensated for the use of their products, and end users have a method to efficiently manage their software assets and verify they are in compliance with the terms and conditions governing the use of those software assets.

The iFOR/LS license management product includes the License Manager Runtime Kit (for clients and servers), the License Management Applications Developer's Kit (ADK) for OS/2, and the License Manager Administration Suite.

The OS/2 License Runtime ADK provides DOS, DOS/Windows, and OS/2 Runtime agents and an OS/2 License Server which will enable license-enabled software to run.

The License Management ADK provides the programming information to allow an ISV to add asset protection to his software program by requesting a license for that program. The APIs, the header files, and the programming documentation are provided with the ADK. Also provided with the ADK without charge is the License Management Runtime agents (DOS, DOS/Windows, and OS/2) and the Runtime Server ADK for OS/2.

The License Management ADK provides the ISV a high level of security that their software products are not being used in an unauthorized manner. The ISV must provide the purchaser a password which contains the number and duration of the licenses granted. Without this, the purchaser will not be able to use the product.

OS/2 License Manager Administration provides a central point at which to remotely administer licensed management servers. The License Management Administrator provides distributed PM object-oriented graphical user interface administration of remote license management servers. This allows:

- Single source of administration of remote license servers
- Automatic software product control
- Central or distributed license control for software packages

```
Gradient Technologies, Inc.
5 Mt. Royal Avenue
Marlboro, MA 01752
(508)624-9600

Contact: Dave Zwicker
```

18.3.8 High Technology Software Corporation

18.3.8.1 ManageView for LAN NetView

HiTecSoft**'s ManageView provides simple commands that allow systems administrators to quickly create utilities to automate network management tasks. ManageView support for OS/2 will integrate with the LAN NetView systems management platform. ManageView also complies with the Network Management Language (NML) specification which guarantees the portability of these utilities across different platforms.

NML is a language specification developed for network management by HiTecSoft NML provides more flexibility for accessing network internals than traditional languages because it is operating-system independent, and because network extensions, such as client/server, distributed processing and smart object architecture, are built into the language.

NML is both powerful and very easy to use. It allows network users to quickly put together network utilities that range from simple programs that will perform repetitive daily routines, to full-blown applications comparable to the ones available commercially. Just as DBMS provides easy commands to manage data, with NML, simple programs can be written that will automate network tasks.

ManageView for LAN NetView is planned to be available by first quarter 1994.

```
HiTecSoft Corp,
3370 N Hayden Rd., Ste 123-175
Scottsdale, AZ  85251

Contact: Sia Moshir
(602)970-1025 x244
(602)970-6323 fax
```

18.3.9 IBM

18.3.9.1 Distributed Storage Manager

Li1.storage management

The ADSTAR* Distributed Storage Manager product from IBM provides storage management and data access services for a large variety of workstation and LAN server platforms operating in a distributed environment. The initial set of functions include policy-based data backup and archive from supported clients to mainframe servers, and the subsequent recovery of that data. The functions may be initiated manually by a client user or scheduled for automatic operation transparent to that user. Other functions include extensive data inventory management and protection, query and reporting, and remote file access support for both bitstream and record-formatted data.

Future releases will continue to expand both client and server platforms, as well as communications protocols, attachments, and function. Scheduled availability is set for first quarter 1994.

```
IBM Corp.
San Jose, California
(408) 284-6387
(408) 284-6725 fax

Contact
Gary Archer
```

18.3.9.2 DatagLANce

DatagLANce* from IBM offers a Token-Ring, Ethernet, and FDDI Network Analyzer. The Token-Ring and Ethernet Analyzer use standard off-the-shelf IBM adapters to perform their network analysis. The following is a list of adapters currently supported:

- Token-Ring
  - IBM Trace and Performance 16/4 Adapter
  - IBM Trace and Performance 16/4 Adapter/A
  - IBM Token-Ring Credit Card Adapter w/TAP Microcode Upgrade
- Ethernet
  - IBM Ethernet LAN Adapter
  - IBM PS/2 Adapter/A for Ethernet Networks
  - Ethernet Credit Card Adapter

DatagLANce Network Analyzer provides:

- A Graphical User Interface
- Ability to monitor the network for statistics on:
  - protocol distribution
  - frame size distribution
  - global traffic statistics
  - special event statistics
  - traffic statistics (supporting text output)
- Ability to capture network frames to a user-definable capture buffer (up to 32MB) with flexible filtering options
- Ability to analyze captured frames with detailed protocol decodes for IBM, TCP/IP, Novell IPX, XNS, DECnet, AppleTalk, and Banyan VINES protocols
- Option to scale the function to whatever platform you wish to use (from a low-cost 286 ISA bus computer to a 50Mhz 486 based Thinkpad 720C --- 25 MHz 386-based platform is recommended).

DatagLANce supports over 103 protocols.

DatagLANce LAN NetView agent will be available in the first half of 1994.

```
DatagLANce
hot-line (919) 254-1364
```

18.3.9.3 LANDP* Product Family

The IBM LANDP product family provides client/server distributed programming capability suited for applications in a LAN environment of heterogeneous operating systems (DOS, OS/2, AIX, and OS/400*) and mixed hardware platforms (Intel, RISC System/6000*, and AS/400). Clients, servers or a combination thereof may reside in any machine. An API, common across platforms, and an open design that supports user-written services is also provided in order to develop distributed services across workstations. OOP language bindings are also provided for application enabling and integration.

The LANDP family of products also provides a significant set of functions, application services and servers to facilitate resource sharing, and communications. Systems management and software distribution via CID enablement are currently available. LANDP systems management capabilities are planned to be integrated with IBM's LAN NetView strategic platform, along with SystemView conformance. This integration will include LANDP agents that will interoperate with the LAN NetView platform.


```
IBM Corporation,
Spain
34-3-40185628562  voice

Contact: Toni Plana
Internet ID: tplana@vnet.ibm.com
```

18.3.9.4 Save Utility/2*

The IBM Save Utility/2 is a full function application for backup and recovery of workstations in a LAN environment. It allows all workstations connected on the LAN to use the centralized store-and-forward capabilities of the system to virtually share the archive device.

Save Utility/2 offers capabilities for:

- Full OS/2 file and operating system backup
- Full DOS file and operating system backup
- Extensive management and administration
- NetBios transport across LAN
- File, subdirectory, disk restoration
- Full system restore from period of last backup
- Enhanced archive device support
- Support of up to 250 workstations per license
- Operates with IBM DOS & OS/2 network products, Microsoft network products (includes Windows), and Novell products

IBM Corp.
Cary, North Carolina
(919) 301-33034
(919) 301-3794 fax

Contact: Scott Pelton

18.3.10 Legato Systems**, Inc.

18.3.10.1 NetWorker**

Legato NetWorker is a powerful, yet easy-to-use backup and recovery product that reliably protects the data on heterogeneous networks.

NetWorker provides high-performance backup and recovery for a wide variety of systems across networks including DOS, DOS/Windows, NetWare servers, and UNIX workstations. Support for OS/2 workstations and the LAN Server platform will be available 1Q94.

NetWorker offers fast backup performance. Key to this performance is NetWorker's ability to back up multiple clients in parallel.

NetWorker provides fully automated, unattended backup. An administrator uses an on-screen calendar to quickly set up a backup schedule of full and incremental backups for each client or groups of clients. Backups can be set to execute at off-peak hours so that system productivity is unaffected.

LAN NetView interoperability is underway for NetWorker, this will allow systems management of Networker clients from an administrator's console.

```
Legato Systems, Inc.
260 Sheridan Avenue
Palo Alto, CA 94306
(415)329-7880
(415)329-8898 fax

Contact: B. Wandel
```

18.3.11 Luxcom, Inc.

18.3.11.1 LC100 Network Manager

Luxcom's OS/2-based LC100 Network Manager provides full monitor and control capability for the Universal Premises Network** (UPN), which integrates data and voice communications using a 100Mps fiber optic isochronous backbone architecture.

UPN offers structured wiring connectivity for:

- 4/16 Mbps Token-Ring and Ethernet LANs
- IBM 3270, 5250, 5080/6090 and RS/6000 terminals
- RS-232/422 asynchronous or synchronous devices
- V.35 equipment

```
Luxcom
3249 Laurelview Court
Fremont, Ca.  94538
Phone: (510) 770-3341

Contact: Roger Biery
```

18.3.12 Microcom, Inc.

18.3.12.1 LANlord for LAN NetView

LANlord is Microcom's systems management solution that specifically addresses the unique requirements of managing personal computers in PC-based LANs. LANlord offers an integrated and complete solution for managing hardware, systems, applications, user and network resources. In addition to the currently supported environments, LANlord will support the IBM LAN NetView platform. Specifically, the following LANlord features are planned to be supported on the LAN NetView platform:

- Hardware and software inventory
- Centralized and remote monitoring of hardware, software, and network resources
  - Statistics and data collection
  - Real-time alerts and exceptions based on predefined thresholds and events
- Software metering
- Reporting and data export

LANlord applications will coexist with other LAN NetView offerings from IBM and other LAN NetView third party management applications.

```
Microcom
1 Executive Blvd, #4
Yonkers, NY 10701
(914)968-2300
(914)968-7100 fax

Contact: Nancy Corelli
```

18.3.13 Network Telesis, Inc.

18.3.13.1 Net-F/X on LAN NetView

Net-F/X provides management with a panoramic view of network performance under real and projected conditions. The system has the unique capability of proactively generating end-to-end traffic and correlating the round trip response times with network statistics to fully describe the effect of varying conditions on remote LAN service levels. The integration of Net-F/X with IBM's LAN NetView extends the system's range from SNA with interconnected LANs to topologies that include SNMP and its suite of protocols. Net-F/X serves a broad range of network management applications because of the centralized control and scalability of its traffic generating facilities. From a LAN NetView platform running the Net-F/X Master Console application, an operator can configure Net-F/X LAN agents to check response times and availability for single user transactions generated to servers and hosts. This technique for remote performance monitoring has distinct advantages over conventional approaches:

- Only one agent per LAN is needed to produce response time reference data.
- Proactive design allows 24 hour 7 day per week connectivity checking.
- Operator specified sample frequency and transaction types.
- Network-wide real-time monitoring.

Net-F/X agents can generate multi-user traffic to hosts and servers proportionate to that produced by applications such as word processing and spreadsheets. Network management can do "live modeling" and realistic production emulation. This sophisticated testing can all be done off hours to avoid adverse effects on production. Net-F/X is completely controllable from a single point which can be located anywhere on the network. Sample intervals, traffic levels, identification of participating agents, data filters, and durations are parameters set from the Net-F/X Master Console. Net-F/X has a useful role in most network applications. The unique Net-F/X traffic generation facility is particularly beneficial in four major areas:

- Performance Analysis.
- Capacity Planning.
- Change Verification.
- Baselining.

LAN NetView strengthens and expands Net-F/X capabilities. Net-F/X on LAN NetView can produce round trip response time reference data for all available paths on an internetwork by using the "discovery functions". Net-F/X can communicate directly with its agents using LAN NetView's SNMP facilities. When needed, a host-based Net-F/X VTAM application program is available to serve as one end of multiple LAN-to-Host sessions to multiplex communications with the Net F/X Master Console application and its remote LAN agents. The LAN NetView SNMP facilities allow Net-F/X to acquire RMON and other LAN statistics for presentation and correlation with response time data and alerts. Performance data is presented at the Net-F/X Master Console in graphical or textual format. The system includes a set of report templates and Net-F/X offers a database utility that simplifies the creation of additional reports. History logs are kept for subsequent trend analysis. Alerts are structured to provide detailed location, time of the event, and response time summary data.

Net-F/X on LAN NetView provides a practical approach to many modern network management challenges. The system can accommodate a range of topologies that makes it compatible with present and future requirements. Net-F/X on LAN

NetView reduces the number of required management tools, saving capital, training and maintenance dollars. It is an easy to use system that presents pertinent information in an understandable manner. Net-F/X can perform its work without adding significant overhead so it runs compatibly with normal production.

Network Telesis, Inc.
27001 Agoura Road, Ste #280
Calabasas Hills, CA 91301-5339
(818)878-7300x110
(818)878-7313 fax

Contact: Roger Mahnke

18.3.14 Novell Inc.

18.3.14.1 NetWare Services Manager v1.5

Novell Inc. has shipped NetWare Services Manager 1.5 (for LAN NetView), an application that provides management of servers running Novell's NetWare network operating system (versions 3.11 and 4.0). NSM 1.5 exploits the power of IBM LAN NetView's management platform to provide users with an integrated offering for managing the Netware servers, while LAN NetView adds applications that provide management of clients, HUBs, databases, and other network resources within the customer's distributed LANs. An administrator can view all these resources from the LAN NetView console.

```
IBM PSP Contact:
1-800 772-2227 (Option 4)
Reference part number 53G6045
```

18.3.15 Opening Technologies**

18.3.15.1 OPENT1**

OPENT1 is a complementary tool to the LAN NetView platform, which is used to help in the development of correct, complete, and consistent managed object definitions. It processes and validates management information object templates, according to the syntax defined by ISO's Guidelines for the Definition of Managed Objects (GDMO). It verifies that all references to other template or ASN.1 definitions are valid. It produces a listing of all processed templates, with any identified syntax errors, a complete cross-reference and listing of the inheritance, naming, and Object Identifier trees. OPENT1 Version 3.0 adds processing of ASN.1 modules, and the production of Managed Object Conformance Statements (MOCS), formatted according to ISO specifications. It provides an API to invoke user-written routines which can generate material such as C or C++ code from the managed object definitions.

```
Opening Technologies
1038 Towlston Road
McLean, Va.  22102
Phone: (703) 759-4647
Fax:   (703) 759-5053

Contact: Jock Embry, Consultant
```

18.3.16 Parallan Computer, Inc.

18.3.16.1 MASS/2**

MASS/2 provides online management capability of the server. It monitors internal temperatures, voltages, and critical server performance conditions. MASS/2 allows the administrator to manage the performance of the server with either local or remote consoles. It also provides users with the capability to set thresholds to generate alarms and alerts. Alerts triggered by thresholds can be reset by the users. Alerts generated by MASS/2 can also be directed to pagers, LANs, and local and remote consoles. The compatibility with IBM LAN NetView is demonstrated through the generation of the alarms and events, which can be directed to a LAN NetView managing system. The MASS/2 MIB extensions will be accessible through LAN NetView metadata services.

```
Parallan Computer, Inc.
201 Ravenwood Drive
Mountain View, Ca.  94034
Phone: (415) 960-0288

Contact: Davis Fields
```

18.3.17 Pragma Systems, Inc.

18.3.17.1 Tower**

Tower is a family of products from Pragma Systems, which runs on the IBM LAN NetView platform. The Tower family provides network and systems management solutions based on both SNMP and CMIP protocol stacks, to manage PCs, servers, bridges, routers, and gateways in an enterprise. The Tower family of products for OS/2 are integrated with the IBM LAN NetView platform and makes use of its topology database and display capabilities to provide a seamless, object-oriented, single focus of management. It also uses metadata information and run-time protocol-based queries to build a dynamic knowledge base so that users can manage objects in a truly browse-and-choose manner.

```
Pragma Systems, Inc.
13706 Research Blvd.
Suite 201
Austin, Tx.  78750
Phone: (512) 219-7270

Contact: Quamrul Islam
```

18.3.18 ProTools Inc.

18.3.18.1 Network Analysis Series:

- **Foundation Manager**
- **Cornerstone Agent**

ProTools Inc. offers two products in its Network Analysis Series which will be integrated into LAN NetView. These products, Foundation Manager and Cornerstone Agent provide powerful, distributed network analysis capabilities in both Token-Ring and Ethernet environments. Foundation Manager is a central console for monitoring and analyzing data from up to 256 remote networks. Cornerstone Agent is an SNMP RMON (Remote Monitoring MIB) agent which also acts as a real-time, stand-alone network monitor. The two products work together to provide distributed network analysis to an organization's internetwork. With the integration, LAN NetView users have the ability to monitor and analyze local or remote networks from a single platform. Foundation Manager can be invoked from the LAN NetView Manage SOM-based User Interface, and it will also share data from LAN NetView's topology/discovery service. The ProTools integration of NAS with LAN NetView will continue to expand over time to include making use of LAN NetView's SNMP communications stack and other standard management and communication functions. NAS for the LAN NetView platform is expected to be available with the availability of LAN NetView.

```
ProTools Inc.
14967 NW Greenbrier Parkway
Beaverton, OR 97006
(503)645-5400
(800)743-4335
(503)645-3577 fax

Contact:  Ellen Recko
Product Marketing Manager
```

18.3.19 Shany Computers, LTD.

18.3.19.1 AlertVIEW

Shany's AlertVIEW is application management software that ensures the smooth operation of end-user applications on any LAN or enterprise-wide network. The AlertVIEW agent monitors applications running on user PCs, detecting existing and potential application errors and gathering detailed information about each one. Relevant data is then automatically reported with standard protocols to both LAN NetView and AlertVIEW's proprietary Management Console.

LAN NetView collects and displays alerts in its standard, familiar manner, while the AlertVIEW Management Console proactively solves most problems, automatically initiating corrective procedures at the user's PC.

```
Shany Computers, LTD
2680 Bayshore Parkway, Suite 104
Mountain View, California  94043

Contact: Tania Sole
(415)694-7410 x242
(415)694-4728 fax
```

18.3.20 Strategic Solutions International Corp.

18.3.20.1 Service Point/32**

Service Point/32 is a fully bi-directional LAN NetView Gateway -- REXX automation Language generates LAN NetView alerts, messages, and expands LAN NetView's systems management over multi-vendor networks.

Service Point/32 provides REXX-based automation applications and a complete development environment. Create standalone or interactive scripts to manage any vendor's element management system from LAN NetView or SP/32. Consolidate and transform alarms and status messages into LAN NetView alerts. REXX language extensions replicate LAN NetView's automation environment on OS/2 including the 3270-like VIEW panels, RUNCMD's, GENALERT's and MSG's. RUNCMD scripts can run in LAN NetView with actual screen images. REXX-based scripts for Alarm Filtering and Alert Generation use database tables. SP/32's datascope displays and captures actual data streams and SP/32's communications simulator provides a complete test environment to minimize development on a production network. Test LAN NetView message automation table logic using simulated network error conditions. Optional C language libraries enable SP/32 functions to be embedded within other applications. Interface with LAN NetView for managing discovered objects. Interface with IBM TCP/IP for SNMP management. Interface with IBM LAN Server for remote LAN server management.

Requires OS/2 2.x and Extended Services Communications Manager or Communications Manager/2.

```
Strategic Solutions International Corp.
1075 Tolland Turnpike
Manchester, CT 06040
(203)649-1900
(203)649-1230 fax

Contact: W. Nathaniel Mills, III
```

18.4 Complimentary Products from IBM

Note: All of the following products in this section are currently available.

18.4.1 System Performance Monitor/2 V2.0

IBM System Performance Monitor/2 V2.0 provides monitoring of OS/2 V2.x critical system resources through a set of integrated data collection, processing and presentation functions. An application programming interface enables the extension of performance monitoring to both 16-bit and 32-bit OS/2 applications. In addition to local data collection, data may also be collected at remote OS/2 LAN Servers or OS/2 LAN Requesters using the SPM/2 distributed feature.

SPM/2 supports a number of recording, logging and graphing utilities to aid analysis and real-time monitoring of a number of system critical resources, which include the following:

- CPU
- Memory
- Files
- FAT Cache
- HPFS Cache
- Physical disk
- Printer
- Communication Port

A follow on version of SPM/2 will be LAN NetView Monitor, on the LAN NetView platform, available fourth quarter 1993.

18.4.2 LAN NetView Start V1.1

LAN NetView Start is a stand-alone product that provides an object-oriented graphical user interface to plan and manage system software and applications on new and existing networks. When used together with CID-based code distribution, LAN NetView Start helps the administrators define network topologies, workstation configurations and LAN connections. REXX control files and validated configuration response files are produced to aid in code distribution across the LAN.

Note that LAN NetView Start is not integrated with the LAN NetView platform.

18.4.3 Distributed Console Access Facility (DCAF)

The Distributed Console Access Facility allows a remote OS/2 workstation to take over the keyboard and screen operations of another remote DOS or OS/2 workstation in the LAN/WAN environment. DCAF can be used to remotely control most full screen text mode and OS/2 presentation manager graphical functions running on an OS/2-based workstation (includes OS/2 Extended Services 1.0), or a DOS workstation on a LAN.

18.4.4 LAN Network Manager V1.1

IBM LAN Network Manager provides powerful LAN media management from either a local workstation or centrally from a NetView host. A Graphical User Interface provides a simple to use view of the network topology, segment or device, using colors to show device status.

IBM LAN Network Manager provides fault, configuration and performance management for Token-Ring LANs, broadband and baseband PC networks, IBM LAN Bridges, and the IBM 8230 Token-Ring Controlled Access Unit (CAU). Function can be further extended to include asset management and access control, by the use of IBM LAN Station Manager. The OS/2 Extended Edition or Extended Services SQL Database Manager is used for building LAN configurations and storing other information such as event logs.

IBM has made a statement of direction that a CMIP proxy agent will be provided in the future to enable an integrated view of the LAN resources on a managed LAN from the LAN NetView platform.

18.4.5 LAN Station Manager* V1.0

IBM LAN Station Manager extends the function of LAN Network Manager by providing workstation-specific information for both DOS and OS/2 LAN-attached stations. Data on workstation hardware configurations and Token-Ring utilization is provided, allowing for the automation of tasks such as asset management. The Common Management Information Protocol (CMIP) is used for information flow to LAN Network Manager.

18.4.6 LAN Management Utilities/2 V2.0

IBM LAN Management Utilities/2 is an OS/2-based set of services to aid in the systems management of LANs and is designed for the client/server enterprise environment. It allows a designated workstation to manage both servers and requesters in an IBM LAN Server and NetWare networks with the following systems management functions:

- Operations Management
- Configuration/Asset Management
- Performance Management
- Fault Management

LAN Management Utilities/2 consists of a graphical display of the LAN, command/data transport, management applications (configuration, performance and fault), and an OS/2 database for data collection. User-written applications and routines can be used to supplement those provided by IBM. While its primary focus is on LAN-attached servers and workstations, it can also provide support for LAN media and transports when used with the LAN Network Manager.

Note: The new name for this product is LAN NetView Management Utilities for OS/2, indicating its migration to the LAN NetView platform. Customers with LAN NetView Management Utilities for OS/2 installed will be provided a migration path which includes keeping the LMU function and integrating it with the LAN NetView family at the user interface. This allows customers to utilize both product's functions during a migration or to keep the LMU products and supplement them with LAN NetView function. This new function in the LMU product allows customers to view LMU as another application which can be added to the LAN NetView family of solutions.

List of Abbreviations

ASN.1	Abstract Syntax Notation.1
CID	Configuration/Installation/Distribution
CMIP	Common Management Information Protocol
CMIS	Common Management Information Services
CMOL	CMIP Over LLC
CMOT	CMIP over TCP
DCAF	Distributed Console Access Facility
DCE	Distributed Computing Environment
DME	Distributed Management Environment
DSOM	Distributed System Object Model
EMS	Event Management Services
FFST	First Failure Support Technology
GDMO	Guidelines for Definition of Managed Objects
HLM	Heterogeneous LAN Management
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
ISV	Independent Software Vendors
ITSO	International Technical Support Organization
LAN	Local Area Network
LMU	LAN NetView Management Utilities
LRF	Local Registration File
MIB	Management Information Base
MOC	Managed Object Catalog
NTS/2	Network Transport Services/2
NvDM/2	NetView Distribution Manager/2
ORS	Object Registration Services
OSF	Open Software Foundation
OSI	Open Systems Interconnect
PCS	Process Control Services
RCLI	Remote Command Line Interface
RMON	Remote Monitoring
ROPS	Remote Operations
SNA	System Network Architecture
SNA/MS	SNA Management Services
SNMP	Simple Network Management Protocol
SOM	System Object Model
TCP/IP	Transmission Control Protocol/Internet Protocol
TMS	Topology Management Services
UDP	User Datagram Protocol
VAR	Value Added Remarketers
WAN	Wide Area Network
XMP	X/Open Management Protocol
XOM	X/Open OSI-Abstract-Data Manipulation

Index

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 948, Building 821
Internal Zip 2834
11400 BURNET ROAD
AUSTIN TX
USA 78758-3493

Fold and Tape **Please do not staple** Fold and Tape

ITSO Technical Bulletin Evaluation RED000

A Collection of IBM LAN NetView White Papers

Publication No. GG24-4207-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction ____

Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

What other topics would you like to see in this Bulletin?

What other Technical Bulletins would you like to see published?

Comments/Suggestions: **(THANK YOU FOR YOUR FEEDBACK!)**

Name

Address

Company or Organization

Phone No.

GG24-4207-00

A Collection of IBM LAN NetView White Papers

GG24-4207-00

PRINTED IN THE U.S.A.

GG24-4207-00