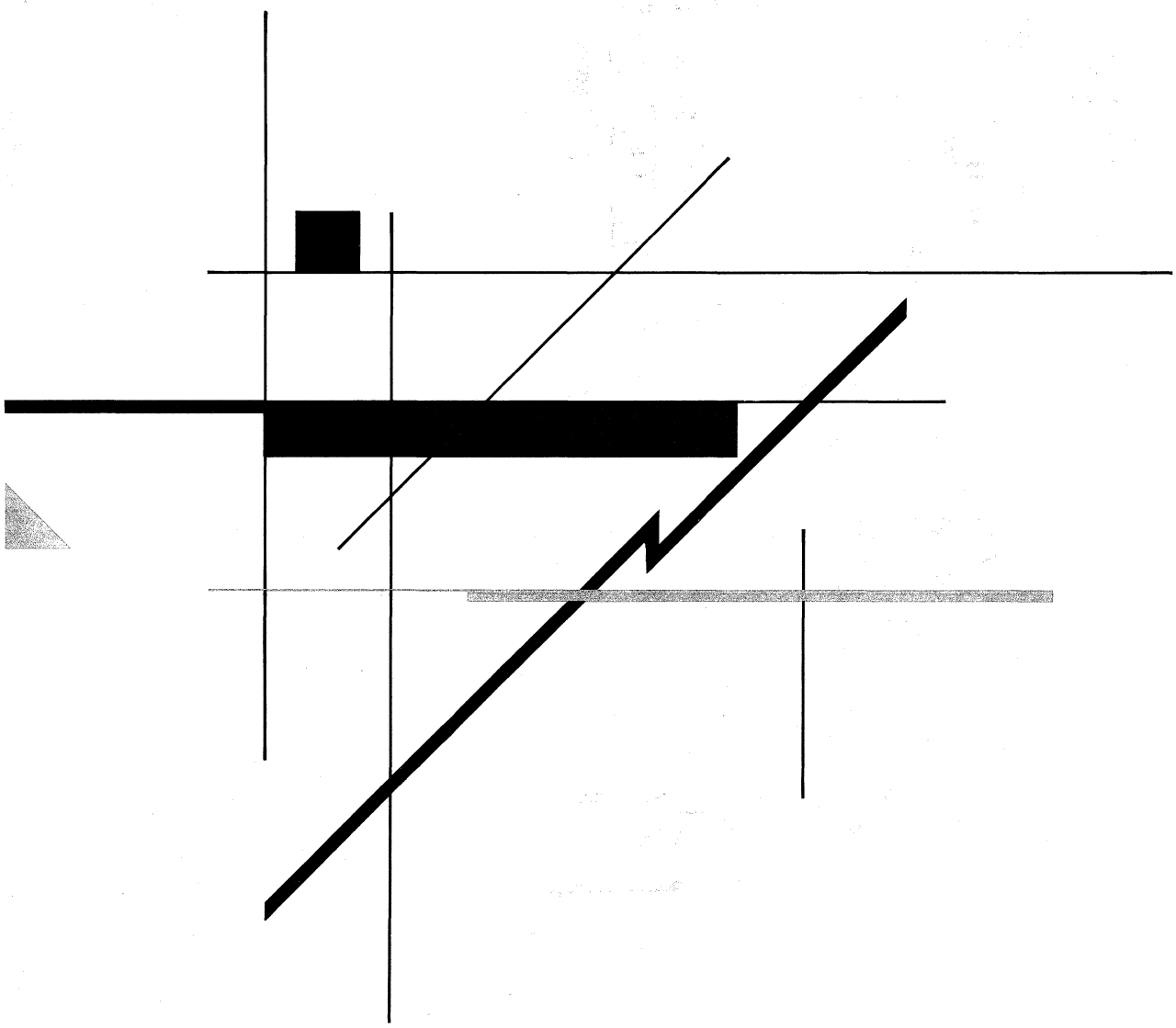
IBM

Systems Network Architecture Management Services

SC30-3346-2

Reference



Systems Network Architecture Management Services

SC30-3346-2

Reference

Third Edition (June 1989)

This edition, SC30-3346-2, is a major revision of the previous edition, SC30-3346-1, and obsoletes that edition; it applies until otherwise indicated in a new edition. Consult Part 3 of the latest edition of *IBM System/370, 30xx, and 4300 Processors — Bibliography*, GC20-0001, for current information on this communication architecture. For a summary of the changes in this book, see "Summary of Changes."

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

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

IBM Corporation
Department E70
P.O. Box 12195
Research Triangle Park, North Carolina 27709, U.S.A.

Special Notices

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 licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used.

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 use these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, New York 10577.

This publication may include references to microcode. Some IBM products contain microcode classified as Licensed Internal Code. Licensed Internal Code is provided under terms and conditions set forth in IBM agreements, such as the Agreement for Purchase of IBM Machines and the Agreement for Lease or Rental of IBM Machines.

IBM is a registered trademark of the International Business Machines Corporation.

Preface

This book provides a comprehensive architectural description of the functions and services associated with Systems Network Architecture/Management Services (SNA/MS). It is intended for systems programmers and program support personnel. It is intended to complement individual product publications. It does not describe product implementations of the architecture.

How This Book is Organized

This book is divided into three parts. It is assumed that the reader of this manual is familiar with the SNA concepts presented in *Systems Network Architecture Concepts and Products*, GC30-3072.

Part I: Introduction to Management Services is introductory information about SNA management services and its categories. An understanding of Chapters 1 and 2 is required before reading the other chapters. Chapters 1 through 7 are organized such that the material can be read straight through. The content is as follows:

- *Chapter 1* introduces:
  - The processes required to plan, organize, and control an SNA network
  - The management services components of a node
  - The choices available to implementations of management services
- *Chapter 2* describes the management services formats and the generic flows that use them.
- *Chapters 3, 4, 5, and 6* describe the management services provided to assist with problem management, performance and accounting management, configuration management, and change management, respectively. Example flows are included.
- *Chapter 7* describes the common operations services available to the individual management services categories. Example flows are included.

Part II: Architectural Logic for Management Services is the detailed description of a model implementation of the management services function sets. It consists of chapters 8 – 10.

It is assumed that the reader of Part II is familiar with Part I. In addition, knowledge of SNA/DS is required. Refer to the list of related publications for more information. Chapters 8 through 10 are organized for ease of reference. The content is as follows:

- *Chapter 8* contains detailed discussions of functions common to a number of management services categories:
  - Transport of management services data using
    - The SSCP-PU session
    - SNA/File Services and SNA/Distribution Services

- How management services identifies resources
- A list of the protocol boundaries that exist between the various management services function sets described in Chapters 9—10.

- *Chapters 9 and 10* provide a concise definition of the management services functions from an implementation perspective. The options and alternatives that implementations may choose are also described.

Part III: Detailed Reference Material contains the following appendixes, as well as the glossary and list of acronyms and abbreviations:

- Appendix A contains Alerts defined for specific environments.
- Appendix B contains management services protocol boundary verbs.
- Appendix C contains SNA/FS file names defined by management services.

The related publications, listed at the end of the preface, are also helpful in understanding the material in this book.

Prerequisite Publications

Systems Network Architecture Concepts and Products, GC30-3072

The following publications are prerequisites for the Change Management and SNA/File Services Support material in Part II:

SNA/Distribution Services Reference, SC30-3098

SNA/File Services Reference, SC31-6807

Related Publications

SNA Formats, GA27-3136

Systems Network Architecture Technical Overview, GC30-3073

Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic, SC30-3112

Token-Ring Network Architecture Reference, SC30-3374

CCITT Recommendation, X.21, 1984

The X.25 - 1984 Interface for Attaching SNA Nodes to Packet- Switched Data Networks, General Information Manual, GA27-3761

IBM 5865/5866 Modems Models 2, 3 Maintenance Information and Parts Catalog, SY33-2048

Systems Application Architecture Common Communication Support Summary, GC31-6810

Relationship to Systems Application Architecture

Not all of the components of SNA/Management Services are included in Systems Application Architecture (SAA). The key components of SNA//Management Services that are currently included in SAA Common Communications Support are:

- Problem management Alerts

Other features of the architecture are included in a range of products. Consult product specifications regarding the status of the feature for the product in which you are interested.

Summary of Changes

The name of this manual has been changed, from *SNA Format and Protocol Reference Manual: Management Services* to *SNA/Management Services Reference.*

The third edition includes new material for the following network management functions:

- The architecture for SNA/Management Services has been extended to an additional category: Change Management.

  Change management capabilities provided to the network planner include planning, scheduling, and tracking of changes to SNA nodes that are typically remote and unattended, during normal operation of those nodes. Functions include Retrieve, Send, Delete, Install, Remove, Accept, and Activate.

  Change management, as described in this document, defines the protocols followed and the formats that flow between nodes implementing this category of the architecture. These descriptions are not intended to address a common user interface or programming interface.

  The Change Management category uses SNA/File Services and SNA/Distribution Services for distribution of potentially large files, requests to manipulate them, and reports to track the distribution and installation.

- A description of the extensions to Query Product Identification (QPI) that comprise the new Network Asset Management function has been added.
- A description of common operations services has been added.
- Alerts have been added to Appendix A for the X.25 environment.

Contents

Part 1. Introduction to Management Services

Part 2. Architectural Logic for Management Services

Part 3. Detailed Reference Material

Figures

Tables

Part 1. Introduction to Management Services

Part I

Chapter 1. Introduction to Management Services

Network Management and Its Major Categories

Network management is the process of planning, organizing, monitoring, and controlling a communication-oriented data processing or information system. The architecture provided to assist in network management of SNA systems is called *management services* and is implemented as a set of functions and services designed to capture and use the information needed for effective management.

This section provides an overall discussion of network management, including those processes for which management services are not provided at present, those that are currently performed manually, and those that are implemented in components outside the SNA node. This general discussion will help the reader understand the use of management services described later in this book.

Network management is divided into the major categories listed in Table 1-1.

Table 1-1. The Major Categories of Network Management
Problem Management
Performance and Accounting Management
Configuration Management
Change Management

Network management processes may be distributed across different nodes, and may require several iterations of data collection and analysis as new events occur. They may be automated system processes or manual processes carried out by network operations or vendor service personnel. Each of the major categories of network management makes use of two types of common services, i.e., services that are implemented once, and then used by each individual category:

- Common operations services: The display and change of system status; automatic presentation of conditions requiring immediate attention; routing of messages among operators, users, and applications; and logging of operator messages (which can be browsed from the operator console).
- Security management: The controlling of safeguards established to protect hardware, software, and data from accidental or malicious modification, destruction, or disclosure.

The individual major categories of network management are described in the following sections.

Problem Management

Problem management is the process of managing a problem from its detection through its final resolution.

Problem is used to describe an error condition resulting in a loss of availability of a system resource to an end user. Problems may originate in hardware, software (operating systems and applications), microcode[1], media, or because of external causes such as user procedures or environmental abnormalities.

Problem management includes the elements listed in Table 1-2.

Table 1-2. The Elements of Problem Management
Problem Determination
Problem Diagnosis
Problem Bypass and Recovery
Problem Resolution
Problem Tracking and Control

SNA management services are provided to assist in performing problem determination and problem diagnosis.

Problem determination is the detection of the loss or impending loss of availability of a system resource to an end user, and completion of the steps necessary for problem diagnosis to begin. It is the process of isolating a problem to the failing hardware device, software product, microcode component, medium, or external cause, to identify the organization responsible for problem diagnosis.

Problem diagnosis is the process of determining the precise cause of a hardware, software, microcode, medium, or externally-caused problem and the precise action required to resolve the problem.

If problem diagnosis is carried out manually, it begins at the end of problem determination. Problem determination output is used to determine the system resource responsible for the problem and the general area where problem diagnosis should begin. If diagnostic data was gathered along with problem determination data, it will be used. If additional data is needed for problem determination or problem diagnosis, it must be gathered and analyzed manually or with the help of SNA management services. In cases of a complex problem, this process may be iterated several times, each time gaining more data and eliminating more components from the set of possible causes.

---

[1] Microcode may be classified as IBM Licensed Internal Code. See "Special Notices" at the beginning of this document for more information.

If problem diagnosis is carried out automatically by a system process, it is usually done in parallel with problem determination, such that the outcome of both processes can be reported together.

Problem bypass and recovery is the process of implementing partial or complete circumvention of a problem, usually before the final resolution of the original incident. This is normally temporary in nature, although it may be permanent. In a simple case, such as printer failure, the system may bypass the reported incident by directing printer output to an alternate printer. In a more complex case, such as controller failure, typically a spare controller must be acquired and placed into service or the existing controller must be repaired, thus substituting problem resolution for bypass and recovery. Problem bypass and recovery frequently takes place in parallel with other elements of problem management; e.g., output may be routed to an alternate printer while the problem with the primary printer is still being diagnosed.

Problem resolution is the process of taking action to correct the error condition detected as a problem or impending problem. This action starts when problem diagnosis is complete and often requires scheduling a repair action, carrying out the repair action, testing the repair, and subsequently reporting the problem as closed and the resource back in service. This procedure is normally the case when a permanent hardware failure occurs. Problem resolution may be simple; for instance, for an inadvertent power-off condition, problem resolution is to turn on the power.

Problem tracking and control is the process of tracking problems until their final resolution. A problem management record is created in the problem data base anytime external intervention is required to restore the system to its proper state of operation. The problem management record provides a repository for all data about a problem to allow correlation with other activities and failures related to the same problem. Some types of this data are problem resolution, status monitoring, and problem status reports.

Performance and Accounting Management

Performance and accounting management is the process of quantifying, measuring, reporting, and controlling the responsiveness, availability, utilization, and usage charges of a network component.

As users become more dependent on networks and network applications, the attainment of acceptable and consistent performance objectives becomes more critical. A poorly performing (or erratically performing) system becomes, in effect, an unavailable system from the end-user perspective. Therefore, awareness of this condition by the party responsible for the management of the information system is required.

Many installation managers establish service agreements or objectives with end users. Typically, these agreements include some response time or availability criteria. Performance management data is required to determine if these agreements or objectives have been met.

Performance and accounting management includes the elements listed in Table 1-3 on page 1-6.

Table 1-3. The Elements of Performance and Accounting Management
Response-Time Monitoring
Availability Monitoring
Utilization Monitoring
Component Delay Monitoring
Performance Tuning
Performance Tracking and Control
Accounting

SNA management services are provided to assist in performing response-time monitoring.

Response-time monitoring is the monitoring of end-user response times and the starting of problem determination if service levels are exceeded.

Availability monitoring is the monitoring of availability and the providing of appropriate data for accounting management.

Utilization monitoring is the monitoring of the utilization of network resources and the starting of problem determination if service levels are exceeded.

Component delay monitoring is the monitoring of the delay incurred at critical components and the starting of problem determination if service levels are exceeded.

Performance tuning is the process of taking action to improve performance. Data available from performance tracking and control is used to identify areas where performance tuning is required.

Performance tracking and control is the tracking and reporting of performance status and the controlling of the effects of tuning actions.

Accounting is the recording and tracking of usage charges at a system resource level. The goal of the accounting function is to provide data to permit proper distribution of the resource cost among the users according to the portion of the system used by each user. The types of accounting data to be collected include connect time, processor cycles used, data quantity transmitted, and class of service for each user session.

Configuration Management

Configuration management is the control of information necessary to identify both physical and logical information system resources and their relationship to one another. The information may include resource names, addresses, location, contacts and phone numbers, vendor or organizations responsible for service, product identification information, and other items. Configuration management assures that this information is updated whenever changes are made and always reflects the current configuration.

Configuration management aids the user in managing the inventory of information system components and assists the other management services categories as follows:

- Problem determination may use the data to determine resource physical identity, location, and possibly the organization responsible for service.
- Change management may use this data to analyze the effect of changes and to schedule changes. Refer to the discussion of change management for the relationship of configuration management and change management.

The following SNA management services assist in configuration management.

Query product identification is the process of retrieving physical identification information (on both hardware and software) from a specified node. The information retrieved is sometimes referred to as *vital product data*.

Change Management

Change management is the planning, control, and application of additions, deletions, and modifications to the information system hardware, microcode and software components. Included are the following:

- Changes to software such as its installation or removal. Software includes system software or application (user) software. Changes can be in the form of fixes, complete load module replacements, or customizing data.
- Changes to microcode, such as its installation or removal. A microcode change could be a patch, a fix, an engineering change (EC), a feature change or customizing data.
- Changes to hardware such as its installation or removal, the application of engineering changes or miscellaneous equipment specification (MES) updates.

The following functions are provided by change management:

- Controlling changes — sending, retrieving, installing, removing, and accepting change files at remote nodes
- Node activation

Change management and configuration management are closely related. Configuration management focuses on the existing configuration of hardware, software, and microcode at any given time. Change management focuses on the planning, application, and tracking of changes to the existing configuration.

Changes occur for two reasons: (1) because user requirements have changed, e.g., for new or additional hardware or a new application system; (2) because a problem requires bypassing or correction. For example, the latter is the case when a failing component is to be removed or replaced to correct a hardware failure, or when an application program module is to be modified to correct a detected program error.

Problems and changes interact with each other in the following manner. While problem bypass or resolution is one of the causes of change, change, in turn, is one of the causes of problems. Problems may be caused by the installation of programs, modifications, or permanent or temporary fixes that were not completely debugged; or by the addition of hardware or software that causes the existing system to perform in a different manner, resulting in a problem that had not been experienced before. Changes that cause problems cannot be completely eliminated, but change management attempts to bring them under control, handle them in an orderly manner, and track them so that problem management can be aware of change activity, if required.

Common Operations Services

Common operations services are a set of services which cross all of the major categories of network management. The architecture for common operations services provides a way to manage types of resources not explicitly addressed by the architecture defined for the individual categories. It does this by providing a general mechanism that allows a network operator to communicate with specialized network management applications; these applications, in turn, provide functions not currently provided by SNA management services.

As an aid in problem management a number of common operation services are provided. They are used in conjunction with any of the management services categories, but are especially useful for problem determination.

- **Execute Command** provides a transmission envelope for any character-coded message or command that is to be executed at a destination node. It provides a means of invoking remotely what would otherwise be a local management services function.
- **Resource Management** services are provided to transport information in an architecturally-defined structure without constraining the content of the information. Query Resource Data and Test Resource collect data and associated identifiers about one or more system or network elements. Test Resource is the more powerful since it requests active testing and returns summary data as well as the detailed data with labels. Analyze Status performs the same type of function but requests an architecturally-defined response rather than resource-dependent information.

Introduction to the Management Services Components of the Node

In an SNA network, the end point of a link or the junction of two or more links is referred to as a *node*. Nodes can be host processors, communication controllers, or workstations. Nodes can vary in routing and other functional capabilities depending on their role in the network. Physically, nodes include the hardware and software components of workstations, controllers, and processors.

SNA Node Types

The role requirements for management services components are categorized according to the type of node in which the component resides. Since the terminology for identifying the various node types within SNA has varied in the past, this introductory section will state the meanings of the terms used in the sections that follow.

system services control point (SSCP) A control point within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network.

subarea node: In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function and boundary function support for peripheral nodes.

type 2.0 node: An SNA node that attaches to a subarea boundary function and receives its management services support via an SSCP-PU session.

boundary-function-attached type 2.1 node: A type 2.1 node that is attached to the subarea boundary function, and receives its management services support via an SSCP-PU session in a manner identical to a type 2.0 node. Since the management services roles for the two node types are identical, this manual will generally not distinguish between the two.

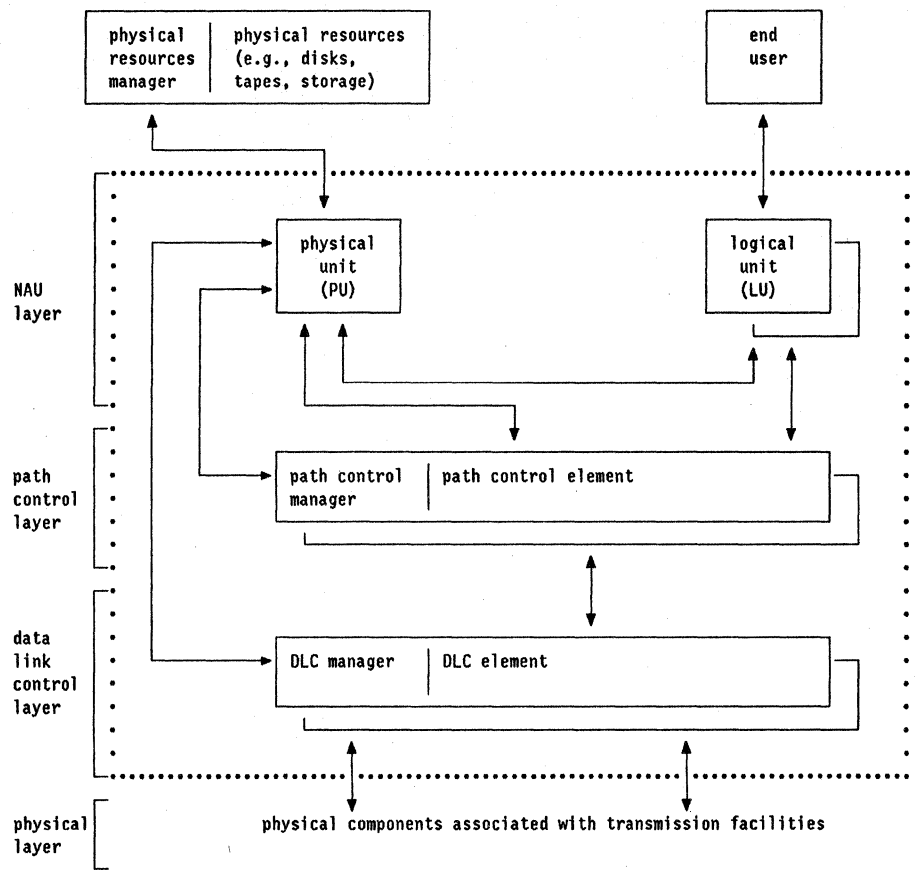


Figure 1-1. Model of an SNA Node (T2.0)

Figure 1-1 is a model of the SNA node type 2.0. Our immediate discussion involves the *network addressable units* (NAUs) — the physical unit, control point, and logical units — within the node. NAUs are sets of SNA components that have names or addresses identifying their routing location, so that they can communicate with one another through the path control network.

Management Services Roles

SNA nodes can be categorized in two basic ways. These notions help the users understand the node's role in the network. An *entry point* is an SNA node that provides distributed network management support. It may be a type 2.0 or type 2.1 node. It sends SNA-formatted network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal-point initiated requests to manage and control its resources.

The concept of a *focal point* was developed to allow the customer the opportunity to centrally manage a distributed network. A focal point is an entry point that provides centralized management and control for other entry points for one or more network management categories.

Focal points and entry points have relationships with each other for one or more categories of network management. Relationships between a focal point and entry points for problem management may or may not be the same as those established for change management, for example. A single communications system or network may have multiple focal points.

The manner in which focal points and entry points interact to accomplish the goal of network management is introduced in "SNA Networks" on page 1-12.

SNA Networks

SNA management services provides operators, whether programmed or human, with the facilities by which an SNA node or a network of nodes may be managed. These include the facilities to manage problems with system resources and to manage system changes, system performance, and system configuration. This section introduces the management services components of a node.

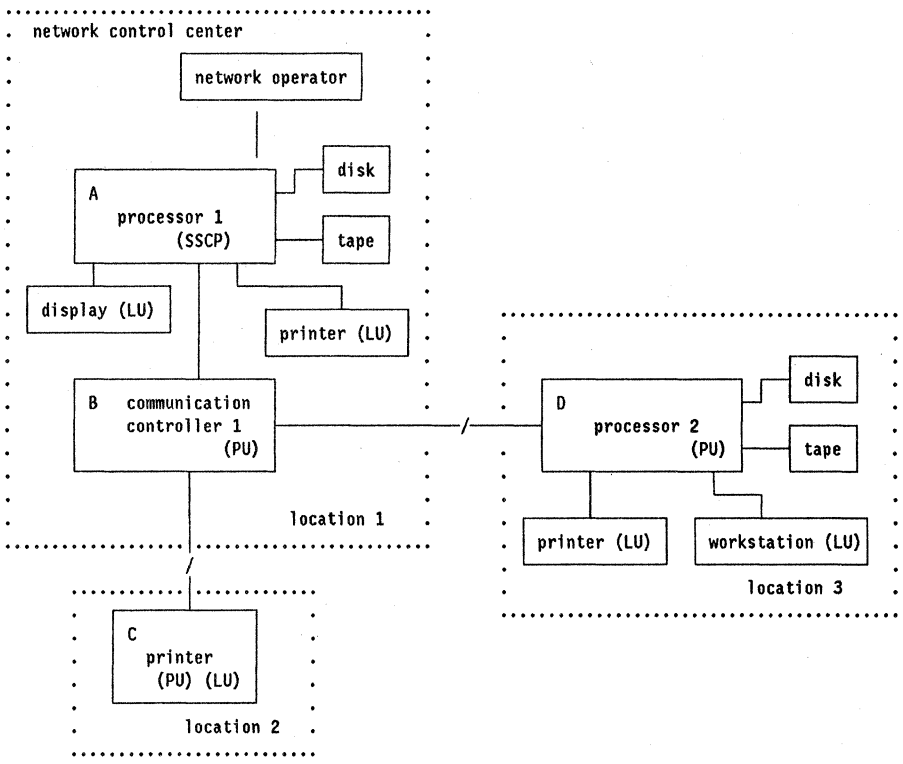


Figure 1-2. Network Example

Figure 1-2 is an example SNA network. In this example, the resources are interconnected with different transmission facilities and media, e.g., channel adapters, local coaxial cable, and links. The nodes are labeled with generic hardware terminology, with the SNA network addressable unit shown in parentheses; a letter is shown in the upper left corner for reference purposes. The term *end user* is used to describe the ultimate source or destination of application data flowing through an SNA network. An end user may be an application program or a person. End users of the network have access to its functions through *logical units* (LUs). LUs allow communication with other end users by establishing sessions between them (i.e., LU-LU session). End users located at displays or workstations have access to network resources at all of the locations.

Each physical location contains at least one SNA node. The *physical unit* (PU) in the node is responsible for managing and monitoring the resources associated with the node. The PU is also responsible for managing links when its node contains the primary link station. For example, the PU at node B (communi-

cation controller 1) is responsible for managing the SNA resources associated with its node (e.g., links as well as logical resources such as sessions and routes) and all of the non-SNA resources (e.g., communication controller storage). The PU at node D (processor 2) is responsible for managing the SNA resources associated with its node (e.g., physical resources such as printers and workstations as well as logical resources such as logical units associated with these printers and workstations) and all of the non-SNA resources (e.g., disk, tape, processor storage).

Node A contains a *system services control point* (SSCP). The role of an SSCP is broader than that of a PU: whereas the PU manages the resources associated with the node in which it resides, the SSCP manages a set of PUs. The term *domain* describes an SSCP and the physical units (PUs), logical units (LUs), links, link stations, and all the associated resources that the SSCP has the ability to control by means of SNA configuration services activation requests and deactivation requests. The SSCP is also responsible for managing the configuration, processing unsolicited data received from physical units, coordinating and routing network operator requests and physical unit replies, and for providing directory support and other session services for end users of the network.

The SSCP may interact with PUs, and a *network operator*, whose job is to manage the network.

Network operators issue commands to, and receive responses from, SSCPs. Many repetitive, supervisory functions use predetermined sequences of commands and responses that can be coded into a program, a technique sometimes described as *automated network operations*. The term *network operator* therefore refers also to any program that manages network operation.

Introduction to Physical Unit Management Services (PUMS)

Physical unit management services (PUMS) is the component of the PU responsible for providing general management services for the node and its associated resources. PUMS communicates with its controlling SSCP using management services RUs transported on an SSCP-PU session and over LU-LU sessions used by SNA/Distribution Services. In a boundary-function-attached type 2.1 node, the CP acts as a PU for the purposes of management services. In general, no distinction is made in this document between type 2.1 nodes in this role and type 2.0 nodes.

The services performed by PUMS are summarized as follows:

- Receiving management services request RUs from a control point with which it has an SSCP-PU session, converting the RUs into requests that are typically implementation-unique, and routing these requests to the appropriate component
- Building and sending RUs
  - Receiving unsolicited management data from other components associated with its node, building an RU, and sending the RU to the resource's controlling SSCPs
  - Receiving solicited management data from other components associated with its node, building an RU, and sending the RU to the control point that requested the data.

  In building the RU, PUMS may reformat implementation-unique data, add other data such as product identification, and assist in correlating requests with replies.

Figure 1-3 on page 1-15 provides a model of the PU and illustrates the components with which PUMS has protocol boundaries. Each protocol boundary is numbered in the figure for subsequent ease of reference. Protocol boundaries exist between PUMS and the following components (numbered to correspond to the numbers in Figure 1-3 on page 1-15).

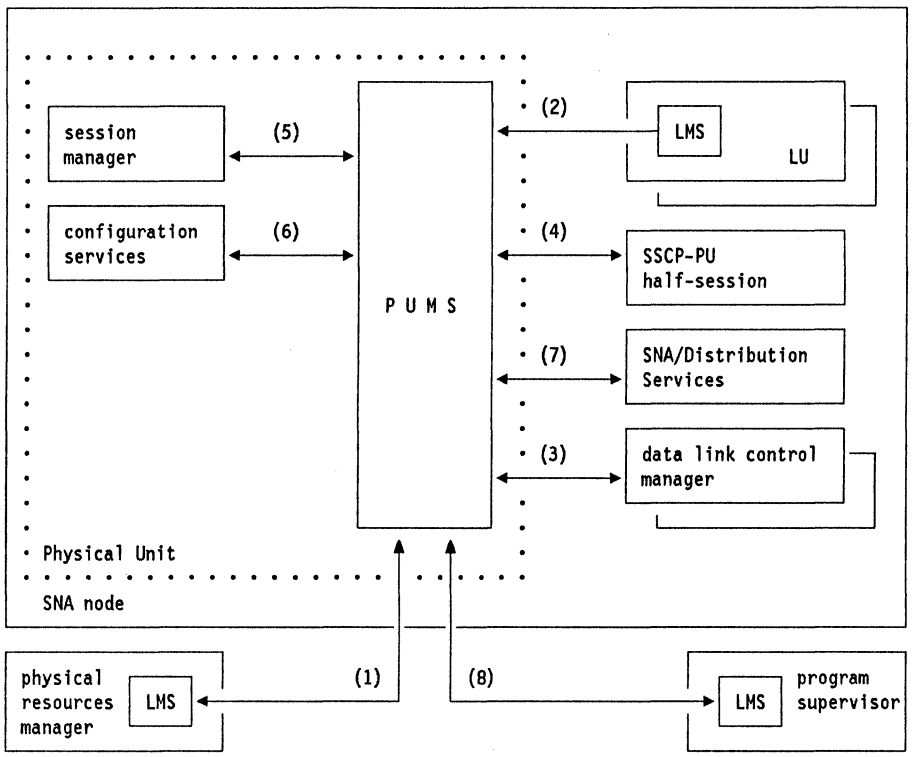


Figure 1-3. PUMS Protocol Boundaries with Other Components

(1) Physical Resources Manager Local Management Services (LMS)

- The physical resources manager LMS provides unsolicited notification of problems with the node's physical resources, e.g., tapes, disks, storage, microcode.

(2) Logical Unit Local Management Services (LMS)

- Upon request, the logical unit LMS sets response-time measurement parameters and provides response-time data.
- The logical unit LMS provides unsolicited notification of problems within the LU and unsolicited response-time data.

(3) Data Link Control Manager Local Management Services

- The DLC manager LMS provides unsolicited notification of problems with links.

(4) SSCP-PU Half-Session

- The half-session provides communication (over SSCP-PU sessions) with a resource's controlling CPMS.

(5) PU Session Manager

- Upon request, the PU session manager provides information about the currently active sessions managed by the PU.

(6) PU Configuration Services

- Upon request, PU configuration services provides information that uniquely identifies the hardware and software of the node, provides a list of active logical units.
- PU configuration services provides unsolicited notification when the SSCP-PU session becomes active.

(7) SNA/Distribution Services (SNA/DS)

- SNA/Distribution Services provides:
  - The capability to send and receive CP-MSUs, SNA/File Services (SNA/FS) agent objects, and SNA/FS files (bulk data) over LU-LU sessions.

The change management category uses SNA/File Services and SNA/Distribution Services for distribution of potentially large files, requests to manipulate them, and reports to track the distribution and installation.

(8) Program Supervisor Local Management Services (LMS)

- Upon request, the program supervisor alters software and microcode components.

Overview of Management Services Communcation

Communication between CPMS and PUMS is accomplished in two ways:

- Over SSCP-PU sessions
- Over LU-LU sessions used by SNA/Distribution Services (SNA/DS)

Communication Between CPMS and PUMS

SSCP-PU Sessions

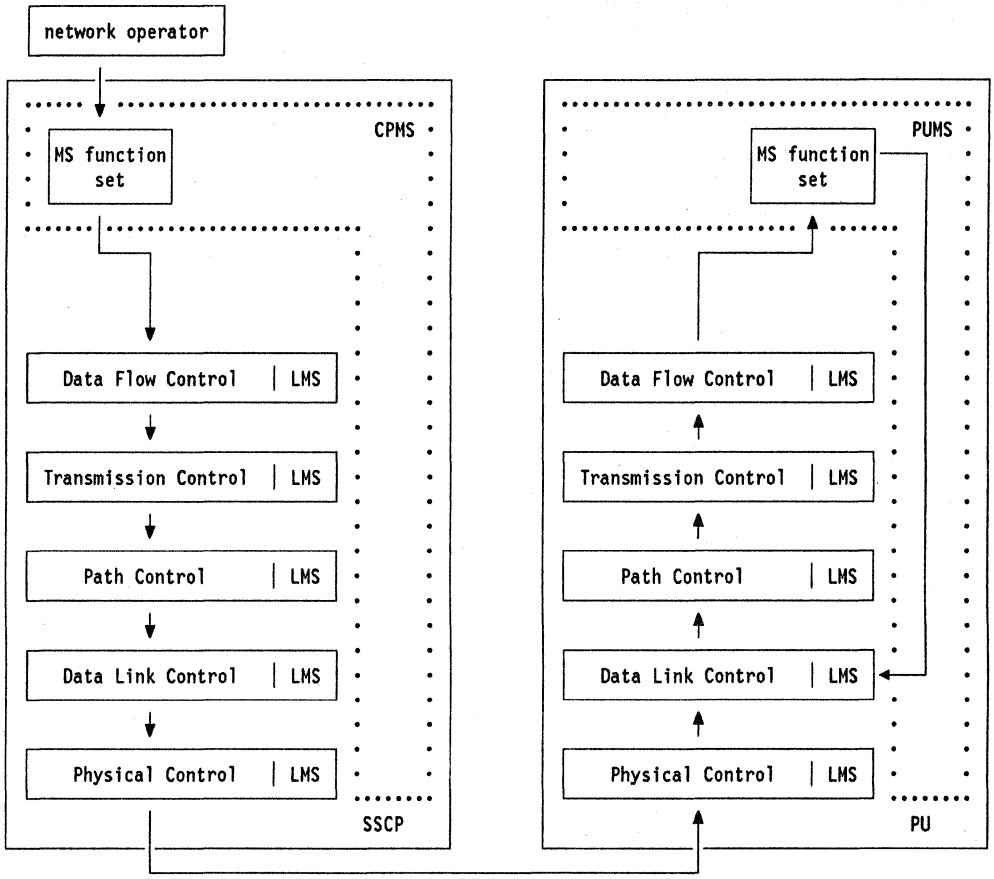


Figure 1-4. CPMS to PUMS Communication on the SSCP-PU Session. PUMS communicates with the LMS for each layer in its node.

Figure 1-4 illustrates how a network operator causes a command to be delivered to a layer manager in a remote node. In this illustration, the network operator interacts with an MS function set at the SSCP controlling the PU. The MS function set prepares the architecturally-defined encoding corresponding to the operator command, then passes it to the data flow control layer (top layer of the SSCP-PU session). The message is passed through the SNA layers at both nodes, and finally delivered by the half-session to the MS function set at the remote PUMS. The MS function set interprets the command and passes the appropriate signals to the *local management services* (LMS) for the layer (data link control in this illustration).

Any replies would follow the same path in reverse to get back to the network operator at the SSCP.

SNA/Distribution Services

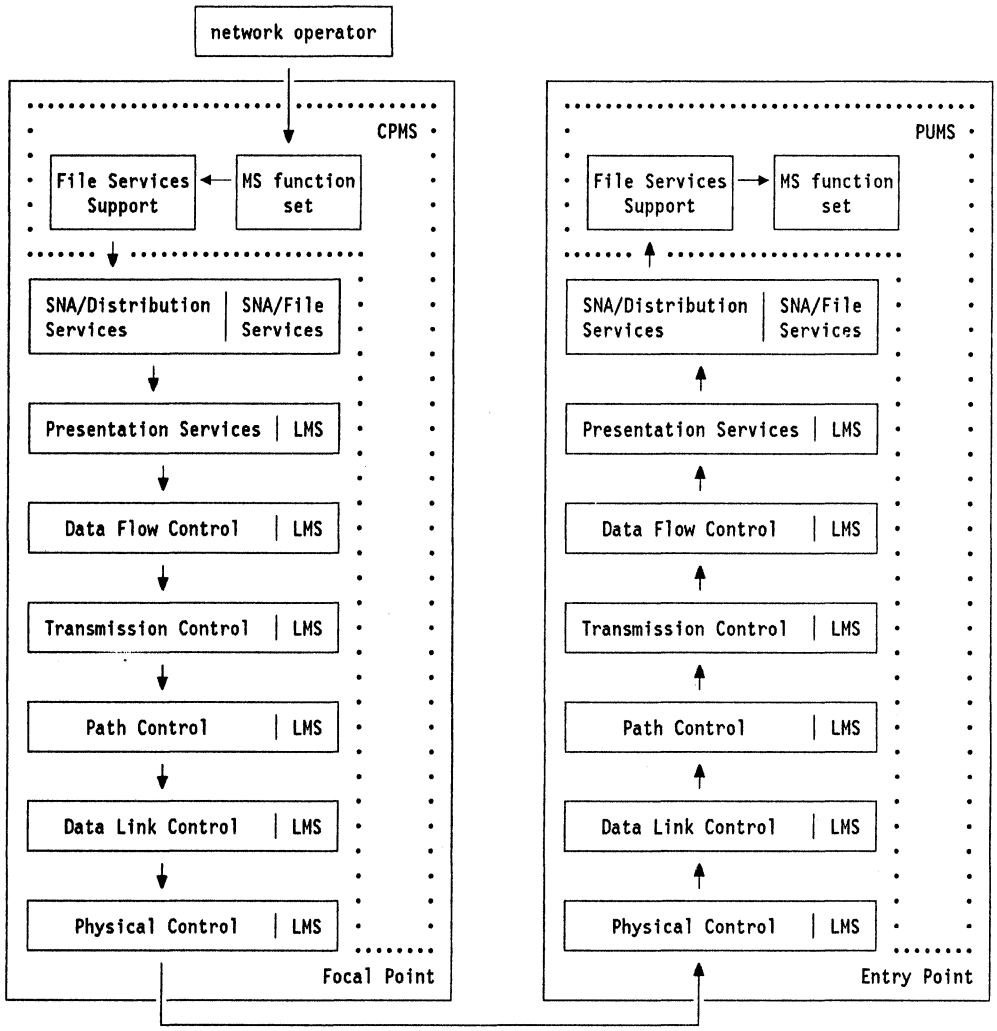


Figure 1-5. Communication Between CPMS and PUMS Using SNA/DS Over LU-LU Session

Figure 1-5 illustrates how MS function sets use SNA/File Services and SNA/Distribution Services for distribution of potentially large files, requests to manipulate them, and reports to track the distribution and installation. SNA/DS uses an LU-LU session, not the SSCP-PU session, for this communication.

This method of communication is used by the change management category.

Implementation Choices

Introduction

Architecture can be implemented in equipment of different capabilities with various physical resources, and with various application specializations. It is not unusual for a given implementation to choose to implement less than the complete architecture. If the various management services functions were selected individually (that is, on a function by function basis), it is unlikely that any two implementations would select exactly the same set, or that the combination of implementations within a network would result in a complete set. As a result, connectivity could be impaired or impossible, and adequate function within a network could not be guaranteed. Therefore, architecture defines the rules under which implementations select the functions to be supported. The rules for selecting the management services functions to implement are divided into the following categories:

- Base and optional subsets of the function sets
- Role requirements
- Electives

Every implementation makes decisions according to the rules in all of the categories. These rules are defined in the remainder of this section.

Base and Optional Subsets of the Function Sets

A management services function set is a collection of services that together perform an overall management services function for a physical unit. In management services, the notion exists of a mandatory or base subset of a management services function set, that all implementations of that particular function set must support. The remainder of the management services function set is composed of options that implementations of that function set may choose to support depending on their role requirements. Some subsets not in the base of one role, may be in the base of another role. So, it is possible to have an "optional subset" that is actually required for a particular role. Other architectures refer to optional subsets as "towers"; this is the same basic concept. The purpose of defining base and optional functions is to allow implementations to implement the set of functions that fits their requirements and to simultaneously enforce a base level of system integrity. The number of options is large enough that, if they were selected on an individual basis, every implementation could have a unique set. This could seriously reduce connectivity. To avoid this possibility, the options are divided into subsets. These subsets are named and numbered.

The composition of a function set is represented by a diagram, as shown in the lower portion of Figure 1-6 on page 1-22. The complete set of functions described by a function set is the outer rectangle. The lower portion of the rectangle contains the subset of functions that is always implemented for that function set (the base subset). The upper portion of the rectangle contains optional subsets that may be chosen to be implemented, depending on role requirements. A subset that is located underneath another subset in the diagram, is a

prerequisite for the upper subset and cannot be omitted unless all the subsets above it are also omitted.

Rules for Function Sets and Their Subsets

- A management services implementation must implement all function sets belonging to the base for its role.
- A management services implementation may implement function sets that are listed as optional for its role, depending on role requirements.
- If a function set is implemented, the base subset of that function set is implemented completely, and optional subsets may be implemented, depending on role requirements.
- Optional subsets of function sets are implemented completely if they are implemented at all.
- An implementation never provides a function defined in a function set in some way different (as perceived outside the node) from that defined by the management services architecture. (An implementation can diverge from the formal model of a node described in this book, but not in ways such that technical variations can be detected outside the node.)

Chapter 9, "General Management Services Function Sets" and Chapter 10, "Specialized Management Services Function Sets for Entry Points" describe the rules for implementation of each of the function sets in further detail.

Role Requirements

Defining a base set of functions that all implementations provide and optional sets of functions that implementations can choose according to their additional requirements is not sufficient to ensure system integrity. If all implementations implemented only a single base set of functions, universal connectivity could be achieved, but some implementations would contain unneeded function while the network as a whole could well be without needed function. Because of this, an implementation's role in the network must be defined before any decision can be made regarding the set of base and optional functions that are to be implemented. A role is composed of base sets of function (base function sets) and optional sets of function (optional function sets). The base subset of each of the base function sets is required to be implemented to define the role, optional function sets may be chosen for implementation depending on product considerations. See the upper portion of Figure 1-6 on page 1-22. The role currently defined for management services is:

- Physical unit management services (PUMS) in a type 2.0 node

A diagram having the general form shown in the upper portion of Figure 1-6 on page 1-22 is used to indicate the functional composition of a particular role. The complete set of functions described by the architecture for that role is bounded by the outer rectangle. The portion of the rectangle below the heavy line (•••••) contains the function sets whose base subsets are always implemented for that role. The portion of the rectangle above the heavy line (•••••) contains function sets that an implementation of that role may choose to include, depending on role requirements. A function set that is located under-

neath another function set in the diagram is a prerequisite for that function set and is not omitted unless all the function sets above it are also omitted. In the example, Role x is composed of function sets a and b and optionally (within the constraints of any stated role requirements), a combination of optional function sets c, d, e, and f. If function set e is selected, function sets d and f are also selected. If function set c is selected, function set d is also selected. Function set d may be selected without function set c, e or f. Function set f may be selected without function set c, d or e.

"Physical Unit Management Services (PUMS) in a Type 2.0 Node" on page 8-21 describes in further detail the rules for implementation of the role defined for management services.

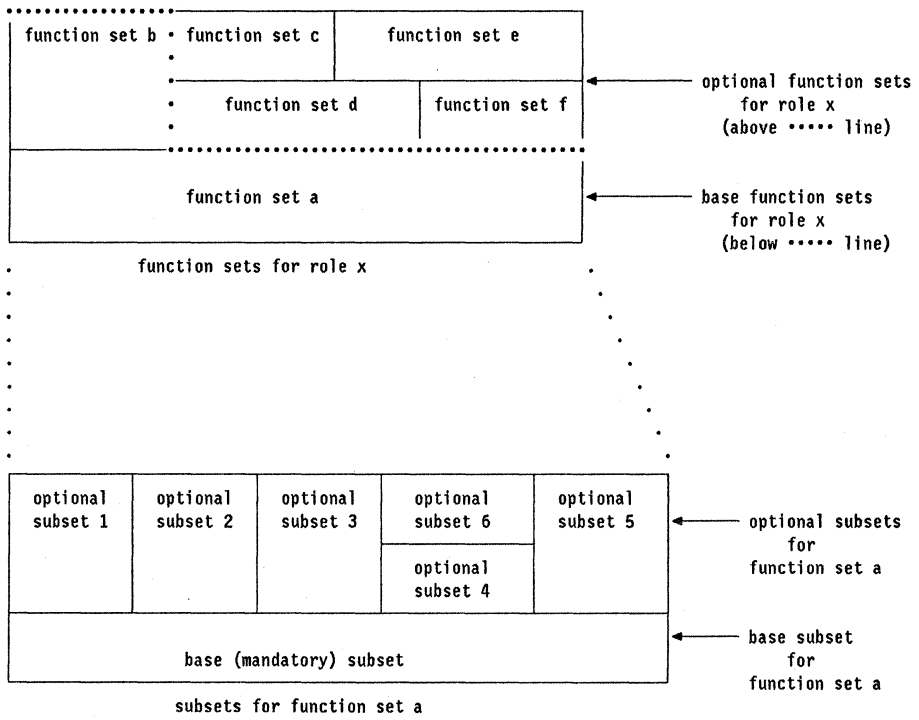


Figure 1-6. Composition of a Role and a Function Set

Electives

Certain functions may be implemented in more than one way. If the effect of the choice is observable outside the management services component of the node, that choice is defined in the architecture as an *elective*. Where the effect of an elective choice is observable by another component, that component must be capable of supporting all the possible effects of the elective choices.

Electives are not optional functions. Optional functions are defined in subsets. Electives are choices as to how or when a function is provided. Implementations make elective choices for performance or development-cost reasons.

Some choices are included as electives because, although they are observable, they cause no, or minimal, inconvenience to the observers while providing significant value to the particular implementations. Other electives do require significant extra flexibility on the part of the other components. In those cases, the development effort saved, or the improved performance provided, are proportionately more important. All electives are documented in the architecture because they have the potential of affecting connectivity.

An example of a management services elective is the method that PUMS uses in reporting that certain requests cannot be processed. The details of why the request could not be processed is described by sense data. PUMS may *elect* to send certain sense data in either a negative response, or a reply. Electives are described within the function sets where they are allowed. Refer to Chapter 10, "Specialized Management Services Function Sets for Entry Points" for additional details.

Chapter 2. Management Services Concepts

Part I

Introduction to Management Services Formats

The overall management services function is provided by the combined functions of the control point management services, physical unit management services, and local management services components. The primary functions of CPMS in an SSCP are to receive requests for management data from network operators and forward these requests to instances of PUMS, and to receive management data from PUMS and forward this data to network operators. The primary functions of PUMS are to receive requests for management data from CPMS and forward these requests to LMS, and to receive management data from LMS and forward this data to CPMS. (Note that CPMS in a type 2.1 node that uses the SSCP-PU session for its management services communication, and performs the same functions as PUMS in a type 2.0 node. It does not perform the functions of CPMS.) The primary functions of LMS are to gather and maintain various types of management data, and to pass this data to PUMS, either in reply to a request from PUMS or unsolicited as the result of a specific event.

Request-response unit (RU) is a generic term for a request unit or a response unit. A request unit is a message unit that contains control information. A response unit is a message unit that acknowledges a request unit; it may contain prefix information received in a request unit. If positive, the response unit may contain additional information (such as route status in response to NMVT), or, if negative, contain the sense data identifying the exception condition.

"Communication Between CPMS and PUMS" on page 1-18 discusses the various communications between CPMS, PUMS and LMS. The following is a summary of those communications.

- A network operator communicates directly with an instance of CPMS.
- CPMS communicates with PUMS on a control point to physical unit (SSCP-PU) session using management services RUs.
- PUMS communicates directly with an instance of LMS.
- Bulk data is transported between CPMS and PUMS using SNA/DS protocols on LU-LU sessions.

The Management Services Formats

Communication between PUMS and CPMS takes two forms. One is communication via management services RUs on an SSCP-PU session. The other is communication via SNA/DS over an LU-LU session. The management services RUs are summarized in Table 2-1.

Table 2-1. Summary of Formats Used by Management Services

SSCP-PU Session	LU-LU Session
NMVT	(X'1212') CP-MSU (X'1532') SNACR (X'1548') FS Action Summary (X'1549') Agent-unit-of-work

The Network Management Vector Transport (NMVT)

The only management services RU used today is the NMVT. Figure 2-1 illustrates the format of the NMVT. Different NS headers identify the different RUs. Externally, the NMVT is no different from any other RUs that flow on this session: its NS header identifies it as an NMVT, and it contains management data. The distinctive feature of an NMVT, the fact that this management data is encoded according to the management services major vector scheme, is of no significance for the NMVT's transport on the SSCP-PU session.

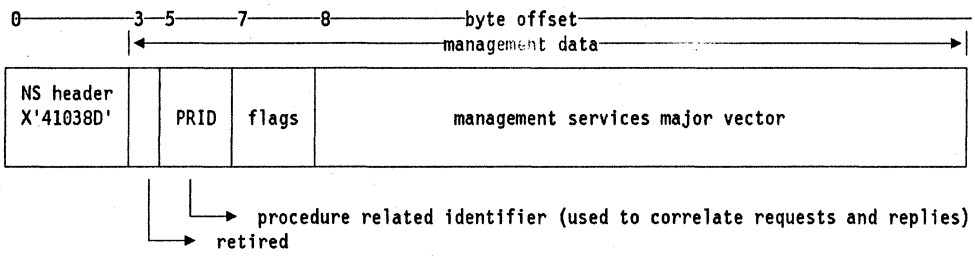
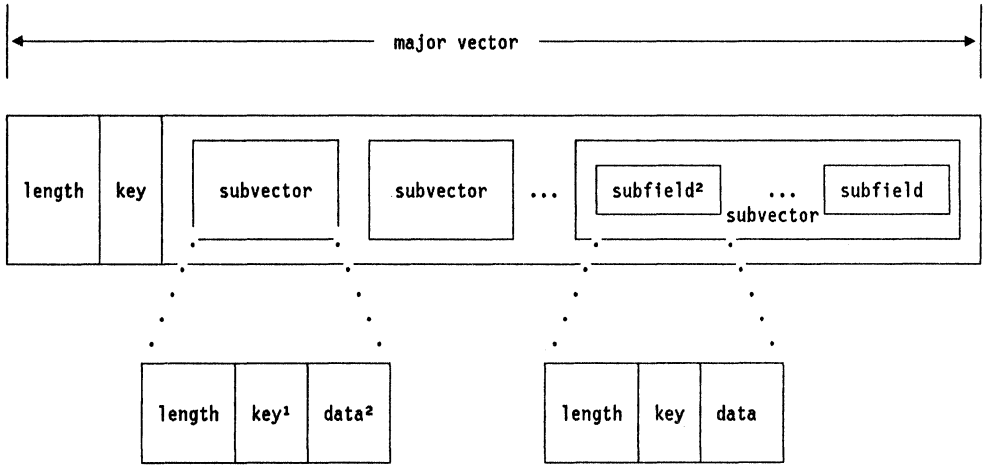


Figure 2-1. Format of the NMVT Management Services RU

Figure 2-2 on page 2-5 provides an overview of the major vector, subvector, subfield encoding scheme of an NMVT. The *key* of the major vector identifies the management function provided. This generally correlates to an element of a major category of network management. For example, the Alert major vector, identified by a key of X'0000', provides the problem determination, and optionally the problem diagnosis, element of problem management. The Response-Time Monitor major vectors, identified by keys of X'0080' and X'8080', provide the response-time measurement element of performance management. The major vector conveys no information other than indicating the function to be provided. It serves only as a carrier of subvectors that carry management services data.

The general encoding scheme for NMVTS is extended in the case of common operations services. Ordinarily an NMVT transports a single major vector. For common operations services, however, one or more *parameter major vectors* are also included, after the regular major vector. Thus the entire structure shown in Figure 2-2 is repeated multiple times within a single NMVT.



[1] Subvector keys are defined as follows:

Subvectors with keys X'00' through X'7F' are referred to as common subvectors. Their meaning is independent of the major vector within which they are used.

Subvectors with keys X'80' through X'FF' have a meaning unique to the MS major vector within which they are used.

[2] Data fields of subvectors may contain subfields.

Figure 2-2. Overview of a Management Services Major Vector

Systems Network Architecture Formats, GA27-3136, documents the NMVT, and the MS major vectors. Each major vector description contains a matrix defining those subvectors that may be contained in it. Those subvectors unique to a major vector are documented after it, while those subvectors that can be used by multiple major vectors are documented at the end of the major vector section, and are called *common subvectors*. Subfields are documented within the subvector in which they are used.

Formats for SNA/DS

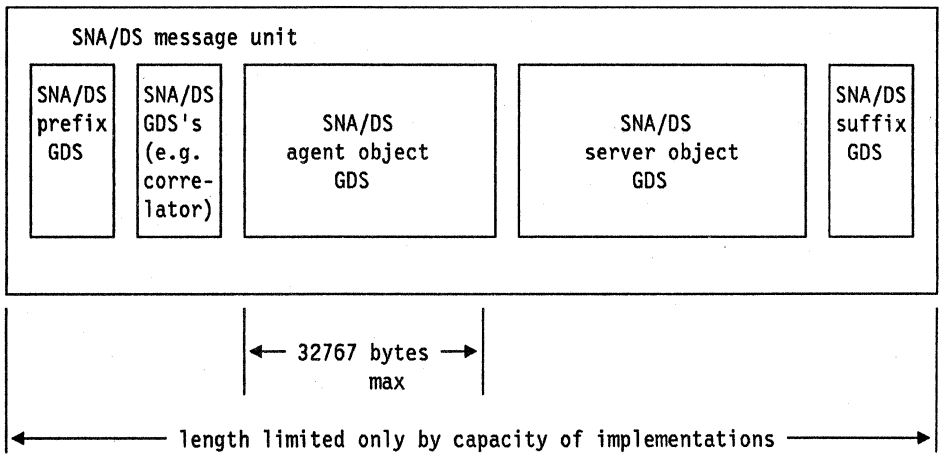


Figure 2-3. Structure of the SNA/DS Message Unit Used by SNA/MS. The SNA/DS message unit contains control information used by SNA/DS, and GDS variables containing application data such as a correlator, an agent object containing agent-to-agent instructions, and a server object containing file control information and a file.

Figure 2-3 is a conceptual view of the encoding used by SNA/DS, the message unit. The message unit contains the following SNA/DS GDS variables:

- SNA/DS prefix and suffix GDS variables to delimit the message unit.
- An SNA/DS agent object, present if agent-to-agent instructions are required by the MS application. For example, an instruction to retrieve a file is carried in the agent object, but when the file is returned no agent object is necessary.

  There are two types of agent-to-agent instructions:

  1. SNA/MS commands or reports, contained within a CP-MSU; and
  2. SNA/FS commands or reports, contained with an SNA/FS GDS variable.

- An SNA/DS server object, present if the MS application requires use of SNA/FS for a particular request. An MS application may use SNA/DS to transport commands and replies that do not require bulk data.

  The server object always contains SNA/FS GDS variables containing file control information, but may or may not contain the file in addition. For example, a request to retrieve a file contains a server object with file control information only.

SNA/DS Message Size:

The maximum size permitted for the agent object is 32767 (X'7FFF') bytes. This is because it is to carry agent-to-agent instructions rather than bulk data. The size of the server object that does carry the bulk data is constrained only by

implementation limits; therefore so is the size of the message unit containing the server object.

Generic Management Services Flows

There are three general patterns for the flow of management services data in a network, termed the *unsolicited* flow, the *request without reply* flow, and the *request/reply* flow. Each management services component, i.e., CPMS, PUMS, and LMS, has a well-defined set of responsibilities for each type of flow. The following sections discuss briefly each of these generic flows.

The following provide a brief discussion of the individual management services flows:

Chapter 3, "Problem Management"
Chapter 4, "Performance and Accounting Management"
Chapter 5, "Configuration Management"
Chapter 6, "Change Management"
Chapter 7, "Common Operations Services"

The responsibilities of PUMS and LMS for each type of flow are described in:

Chapter 9, "General Management Services Function Sets"
Chapter 10, "Specialized Management Services Function Sets for Entry Points"

The figures referenced in the next sections have numbers along the left side identifying each stage of the flow. These same numbers are used in the description of each stage.

Unsolicited Flows

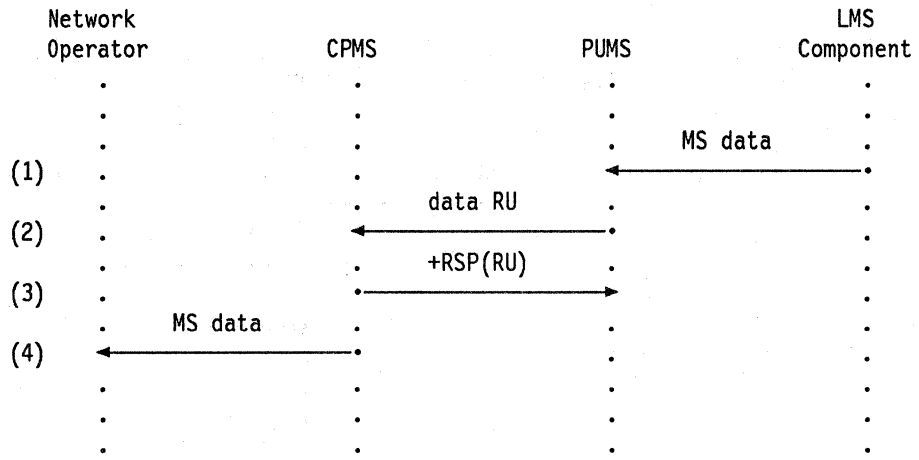


Figure 2-4. The Unsolicited Flow

Figure 2-4 shows an unsolicited flow of management services data. The flow consists of the following stages:

1. As a result of the occurrence of some specified condition, an LMS component passes management services data to PUMS. An example of a condition that initiates an unsolicited flow is detection of an error of a certain type. An exhaustive list of the conditions under which an LMS component passes data unsolicited to PUMS appears in the discussion of each individual management services function.
2. PUMS formats the data it receives into a management services RU, in some cases adding additional data of its own such as product identification; then it sends the RU to CPMS on an SSCP-PU session. The general term "data record," and the more specific "data NMVT," refer to management services records flowing from PUMS to CPMS. The terms apply to both solicited and unsolicited records. A management services record flowing in the other direction, from CPMS to PUMS, is termed a "request NMVT."
3. CPMS sends a response upon receiving the RU.
4. CPMS passes the management services data to the network operator. Depending on the type of data involved, additional functions may be provided, such as routing the data to different operators, logging the RU, and analyzing the data in certain ways.

Request Without Reply Flows

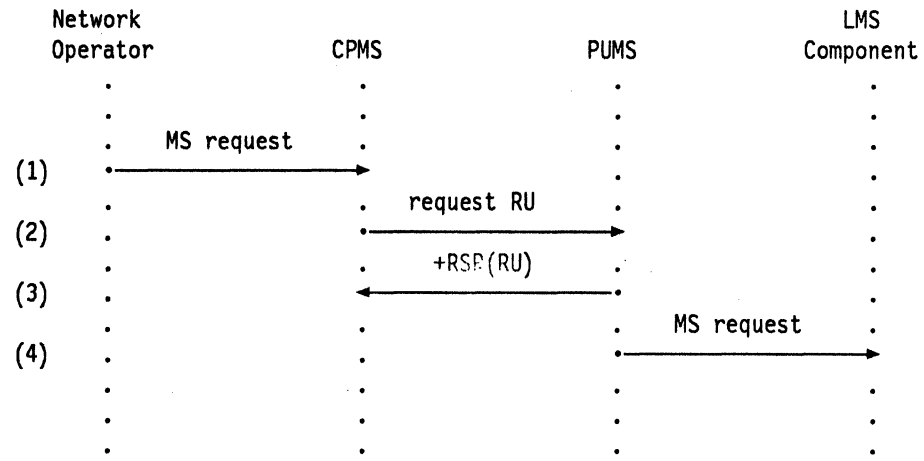


Figure 2-5. The Request Without Reply Flow

Figure 2-5 shows a management services request, typically a command, for which no reply is expected. The flow consists of the following stages:

1. A network operator enters a management services request for one or more specified resources. Certain requests allow the operator to specify only a single resource; others allow for multiple resources to be specified implicitly, by, for example, directing a command to *all* LUs associated with a specified PU. In any case, all resources specified in a request must be associated with a single PU. See the discussion of the individual flow for statements of the types of requests supported. The network operator's request is passed to CPMS.
2. CPMS reformats the request and sends it to the appropriate instance of PUMS on an SSCP-PU session.
3. PUMS receives the request RU and parses it. If it finds an error, it sends a -RSP to CPMS. Otherwise, it sends a +RSP.
4. Having received a management services request, PUMS passes the request to the appropriate LMS component. If the request RU specified more than one resource, PUMS passes multiple requests, one for each resource to which the request was directed.

Request/Reply Flows for Non-Bulk Data

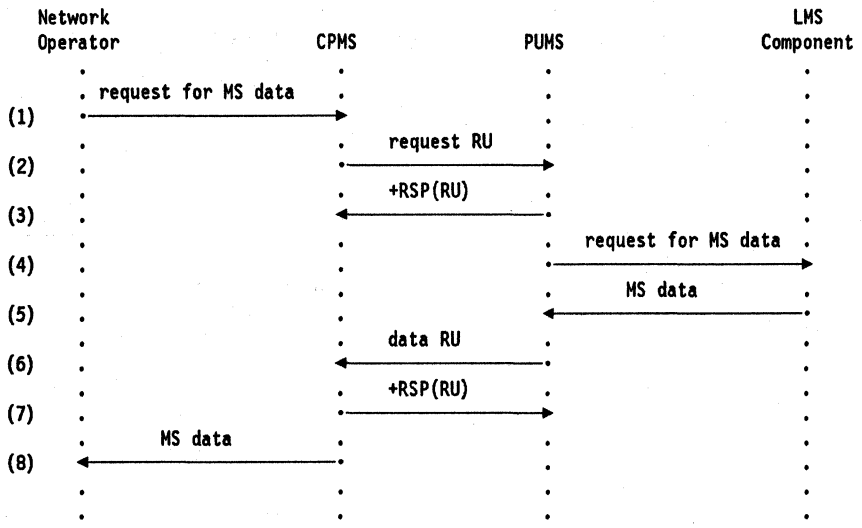
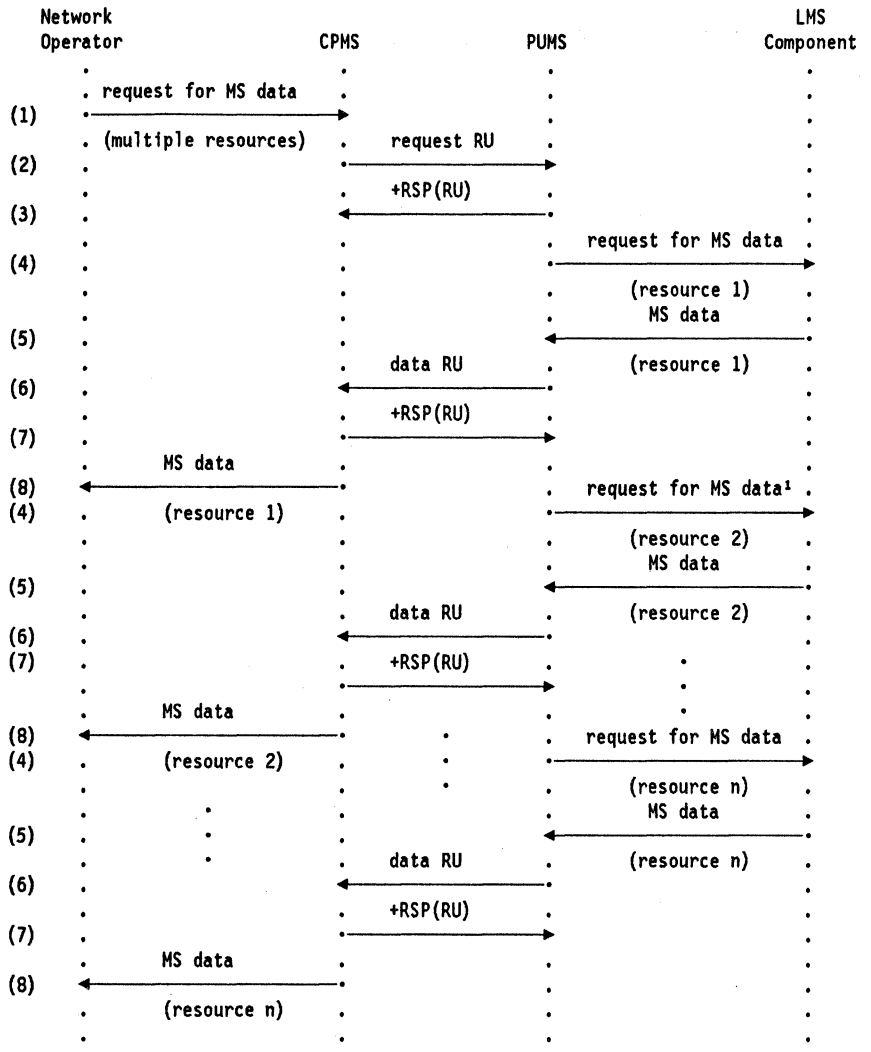


Figure 2-6. The Request/Reply Flow for Non-Bulk Data



[1] The architecture allows requests to the LMS to be made before previous requests have been replied to and sent to CPMS.

Figure 2-7. The Request/Reply Flow for Non-Bulk Data (Multiple Resources). This flow is the same as the flow for a single resource (Figure 2-6) except that stages 4 through 8 are repeated for each resource specified.

Figure 2-6 on page 2-10 and Figure 2-7 show two varieties of a request/reply flow for non-bulk management services data. The flow consists of the following stages:

1. A network operator requests a particular type of management services data from one or more specified resources. Requests for certain types of data allow the operator to specify only a single resource; others allow for multiple resources to be specified implicitly, by, for example, requesting data from *all* LUs associated with a specified PU. In any case, all resources specified in a request must be associated with a single PU. See the discussion of the individual flow for statements of the types of requests supported. The network operator's request is passed to CPMS.

2. CPMS reformats the request and sends it to the appropriate instance of PUMS on an SSCP-PU session.
3. PUMS receives the request RU and parses it. If it finds an error, it sends a -RSP to CPMS. Otherwise, it sends a +RSP.
4. Having received a request for management services data, PUMS passes the request to the appropriate LMS component. If the request RU specified more than one resource, PUMS passes multiple requests, one for each resource for which data was requested.
5. The LMS component acts on each request that has been passed to it. It gathers the requested data and passes it to PUMS.
6. PUMS builds a management services RU with the data that was passed to it, including additional data of its own, such as a time stamp; then it sends the RU to CPMS on an SSCP-PU session.
7. CPMS sends a response upon receiving the RU.
8. CPMS passes the requested management services data to the network operator that made the request.

One aspect of the soliciting/solicited flow not apparent in Figure 2-6 on page 2-10 and Figure 2-7 on page 2-11 is the multiplicity of levels at which request/reply correlation must take place. PUMS must first correlate the data it receives from LMS with the request that elicited it. Next, CPMS must correlate the data RUs it receives with the request RUs that it has sent.

Correlation is made more complicated in all of these instances by the following:

- MS major vectors may be sent either solicited or unsolicited. Thus, in a particular case, one of these major vectors may correlate with no request at all.
- When multiple resources are specified on a request, more than one reply will correlate with that single request. Thus, at both points of correlation, it must be determined not only which request a particular reply correlates to, but also when the *last* reply for a given request has been received.

The NMVT contains a Procedure Related Identifier (PRID) field to be used by the CPMS to correlate the data RUs it receives with the request RUs it sends. Chapter 10, "Specialized Management Services Function Sets for Entry Points" provides further details of how request/reply correlation is done at each level.

Request/Reply Flows for Bulk Data

Retrieving Bulk Data

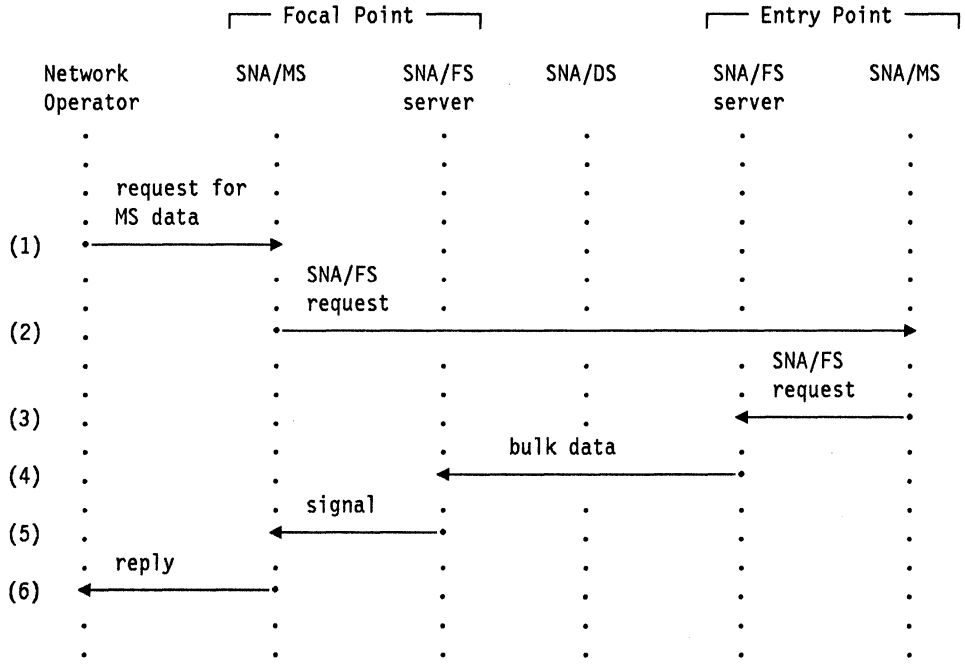


Figure 2-8. The Request/Reply Flow for Retrieving Bulk Data from an Entry Point

Figure 2-8 shows a request/reply flow to *retrieve bulk* management services data.

SNA/File Services (SNA/FS) architecture is used by management services for bulk data transport. For a description of this architecture, refer to *SNA/File Services Reference, SC31-6807*

For a more detailed understanding of the architectural components traversed in each node, and their sequence, refer to "Transport of Bulk Management Services Data" on page 8-4.

The flow consists of the following stages:

1. A network operator requests a particular type of bulk management services data. The network operator's request is passed to SNA/MS in the focal point.
2. Based on the information passed to it in the request, the focal point SNA/MS formats an SNA/FS request. The request flows on a SNA/DS conversation between the focal point and the entry point.
3. Having received a request for management services data, SNA/MS passes the request to the SNA/FS server.
4. The SNA/FS server fetches the requested bulk data and sends it to the focal point SNA/MS on a SNA/DS conversation.

5. The SNA/FS server at the focal point stores the bulk data and SNA/DS notifies SNA/MS that it has arrived, passing the report in the agent object.
6. SNA/MS notifies the network operator that the bulk data has arrived.

Sending Bulk Data

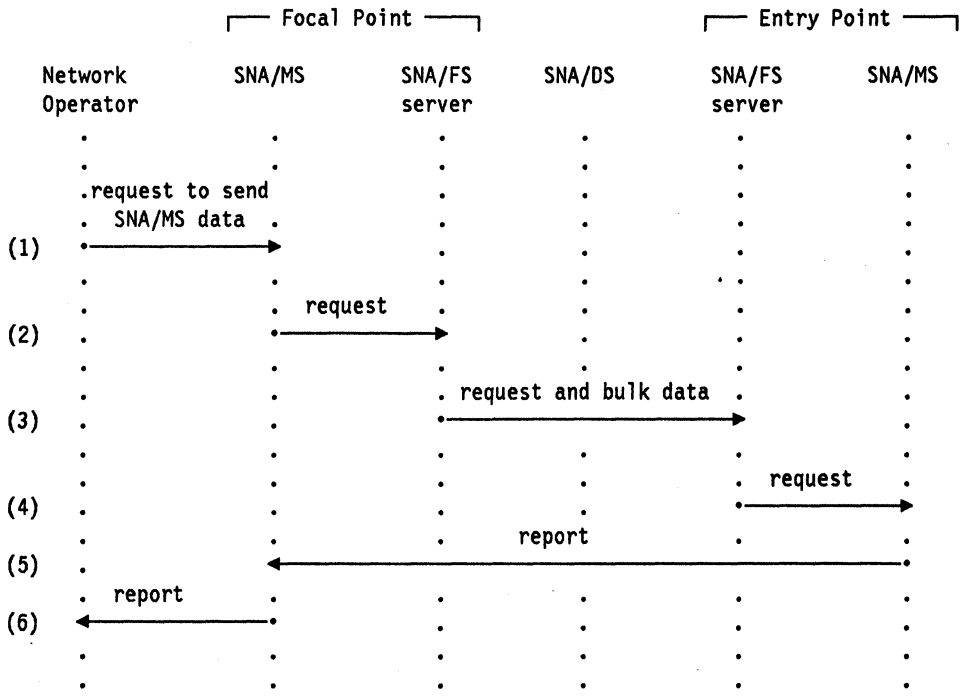


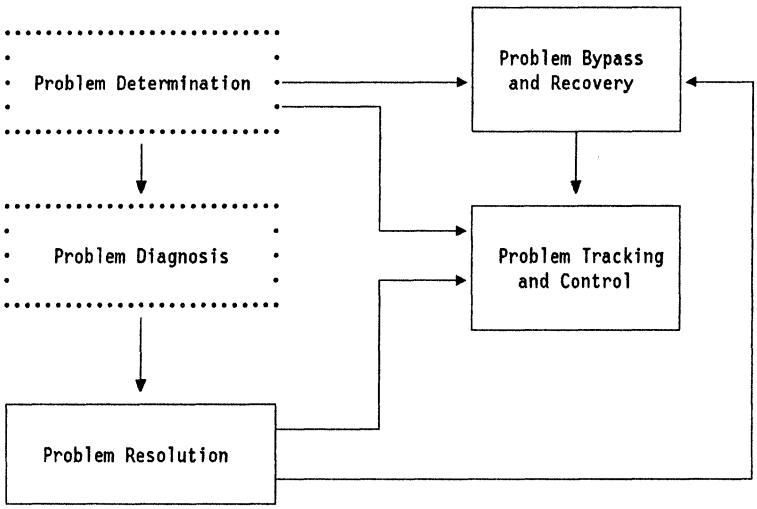
Figure 2-9. The Request/Reply Flow for Sending Bulk Data to an Entry Point

Figure 2-9 shows a request/reply to *send bulk* management services data. The flow consists of the following stages:

1. A network operator requests that a particular type of bulk management services data be sent to an entry point. A management services function related to the data may be requested in addition. The network operator's request is passed to SNA/MS in the focal point.
2. Based on the information passed to it in the request, the focal point SNA/MS formats either a request CP-MSU or an SNA/FS request, depending on whether or not a management services function over and above simple file transfer is requested. SNA/MS invokes its SNA/FS server to fetch the bulk data from storage.
3. The request and bulk data flow on a SNA/DS conversation between the focal point and the entry point.
4. The SNA/FS server in the entry point stores the bulk data and SNA/DS notifies SNA/MS that it has arrived, passing the CP-MSU or SNA/FS request in the agent object.
5. SNA/MS generates either a report CP-MSU or an SNA/FS report as appropriate, and sends it to the focal point SNA/MS on a SNA/DS conversation.
6. After the report has reached the focal point, SNA/MS passes it to the network operator that made the request.

Chapter 3. Problem Management

Problem management is the process of managing a problem or potential problem from its detection through its final resolution. The term *problem* denotes an error condition resulting in a loss of availability of a system resource that is visible to an end user. Problems may originate in hardware, software (operating systems and applications), microcode media, or as a result of external causes such as user procedures or environmental abnormalities. For an overview of the elements of problem management and their relationships to one another, refer to Figure 3-1.



NOTE: SNA management services are provided to assist with elements bounded by •••••.

Figure 3-1. Overview of Flow Between Problem Management Elements

Problem Determination

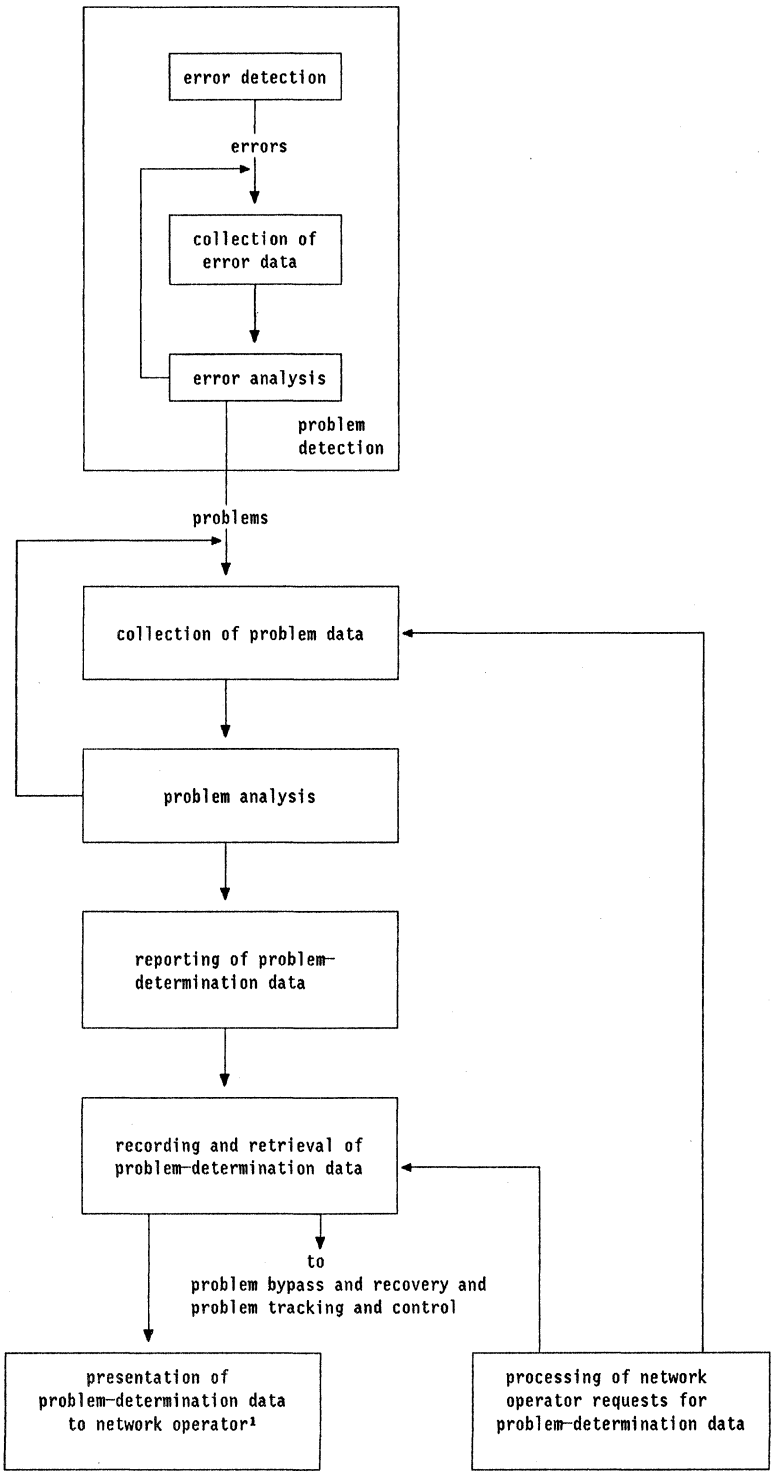
Problem determination is the element of problem management that detects a problem or impending problem and completes the steps necessary for problem diagnosis to begin. Refer to Figure 3-2 on page 3-5 for an overview of the steps required for problem determination.

The problem-determination process may be performed automatically by a system process, may be performed as a series of manual processes, or may be a combination of both automated and manual processes. There are two types of problem determination: node problem determination and system problem determination.

- Node problem determination addresses the class of problems associated with a node and its resources that are managed by a local node operator (either programmed or human). Typical of the type of problem managed by a local node human operator are "intervention required" situations such as "out of forms," "forms jam," and "mount requests."

- System problem determination is managed by a network operator at a network control center, and provides the management of those problems that are not managed by the local node operator.

The term *problem determination* as used in this book refers only to system problem determination.



[1] The network operator will contact personnel responsible for problem diagnosis.

Figure 3-2. Overview of Problem-Determination Steps

Overview

To complete system problem determination, network operators (human or programmed) must first be aware that a problem exists and then have access to the data that will lead to the identity of the failing system resource, e.g., a machine or a link, and the data required to determine the organization responsible for its maintenance. System problem determination is composed of the following steps:

Problem Detection

The detection step of problem determination provides for the recognition of the loss or impending loss of availability of a system resource to an end user. The action of a network operator may be required for some or all of the problem determination, problem diagnosis, or problem resolution processes. The following conditions are detectable:

- Permanent loss of the availability of a resource that can be corrected only by intervention external to the reporting component
- Temporary loss of the availability of a resource that is corrected without intervention external to the reporting component

  This includes cases where the resource cannot consistently be used in the manner in which it was intended because of recurring errors.
- Response-time degradation beyond a predetermined value
- Problems recovered from, the existence of which prevented reporting at the time of occurrence
- Anomalies that have the potential to cause an interruption in availability to the end user, e.g., display screens starting to distort, a temperature out of a tolerable range, or a machine emitting a strange noise

Problem detection may be performed by any of the following:

- Local management services
- An end user or node operator (in cases where a problem is not automatically detected by local management services)

Collection and Analysis of Problem Data

The collection and analysis step of problem determination determines the following:

- Problem origin and cause
  - Physical identity of the system resource that caused the problem; e.g., machine type, model number, and serial number
  - Logical identity of the system resource that caused the problem, e.g., the name or address used by the system to identify the resource (This information is dynamic and may be changed by system generation or customization options.)
  - A very brief generic description of the problem and classification of its cause to one of the following:

  - Hardware
  - Software
  - Microcode
  - Media
  - External cause; e.g., the end user

  This information is used in conjunction with system and/or user procedures so that the service organization responsible for problem diagnosis can be determined, directed to the location where the problem exists, and have a brief description of the problem to be worked on.

  Sufficient data to complete problem determination should be collected at the first occurrence of a problem. When possible, problem diagnosis data is also collected and analyzed at this time. If sufficient data is not collected at the time of the initial error, the problem will have to be re-created before problem determination and/or problem diagnosis can be completed. This is often unreliable and time consuming if the problem is intermittent.

  The analysis step processes not only problem data (as collected by the problem-data collection step) but also other network management data that may be pertinent to the problem, such as change management or configuration management data.

- Action to be taken by the network operator to complete problem determination
- Severity or impact of the problem; e.g., permanent or temporary, so the network operator may set the priorities of the actions required
- Correlation of the following:
  - Data related to this problem that is required for problem diagnosis or problem resolution
  - End-user verbal reports of this same problem
  - Other unsolicited notifications for this same problem sent by other nodes; Alerts are an example of of unsolicited notifications which are explored further in this chapter.

The collection and analysis of problem data may be performed by some or all of the following:

- The detector of the problem, i.e., the end user or local management services
- The local system operator (if applicable) if analysis had not been completed when the problem was reported by either an unsolicited notification or a verbal report
- The network operator if collection and analysis had not been completed when the problem was reported by either an unsolicited notification or a verbal report

  The collection may be an iterative process and may use any or all of the following methods:

  - Manually observing other indications at the control point

- Requesting the local node operator at the node reporting the problem to manually observe other indications of the problem at the node
- Requesting the local operator at the node reporting the problem to use services available at that node

Reporting of Problem-Determination Data

The reporting step of problem determination provides for reporting of problem-determination data to the control point responsible for managing the resource determined to be the origin of the problem.

- If an end user detected the problem, it may be reported directly to the network operator by a phone call or to the control point by a manually-initiated management services RU (Alert). It is optional for products to provide a mechanism to allow an operator or end user to initiate a management services RU (Alert).
- If local management services detected the problem, it would report it to PUMS to be reported to the control point using a management services RU (Alert).

Recording and Retrieval of Problem-Determination Data

The recording and retrieval step of problem determination is implemented at the control point, processes only that data reported via Alerts, and provides the following:

- Selective logging of Alert data on the Alert data base, based on criteria specified by the network operator
- Data base management (e.g., storage, retrieval) of the collected data
- Interaction with the problem tracking and control and problem bypass and recovery elements

Presentation of Problem-Determination Data

The presentation step of problem determination is implemented at the control point and provides the following:

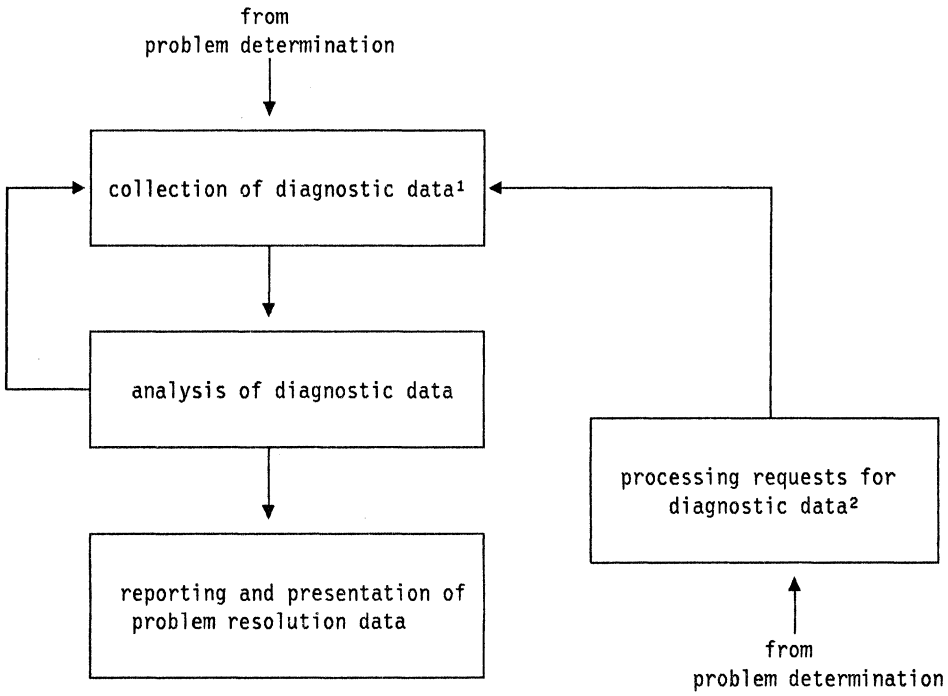
- Selective presentation of Alert data to the network operator based on criteria such as Alert type or product type
- Formatting and presentation of problem-determination data to the network operator in a manner designed to foster real-time awareness

Processing of Requests for Problem-Determination Data

The request step of problem determination is implemented at the control point and provides a mechanism for the network operator to request data from the Alert data base, or from the data-collection step for those cases where the data was not automatically reported.

Problem Diagnosis

Problem diagnosis is the element of problem management that determines the precise cause of a problem and reports the precise action required to resolve the problem. Figure 3-3 gives an overview of the steps required for problem diagnosis.



1 If this process is performed automatically, it is started by the problem detection function of problem determination.
2 If this process is not performed manually, the network operator will contact the personnel responsible for controlling it.

Figure 3-3. Overview of Problem Diagnosis Steps

The problem-diagnosis process may begin at the node detecting the problem and may be distributed through as many other nodes as is necessary for completion. The process may be performed automatically by a system process, may be performed as a series of manual processes, or may be a combination of both automated and manual processes.

If problem diagnosis is performed automatically by the system, it is usually done in parallel with problem determination, beginning with the collection step and ending with the presentation of both problem-determination and problem-resolution data. See Figure 3-4 on page 3-16 and Figure 3-5 on page 3-17 for details.

If problem diagnosis is performed manually, it begins at the conclusion of problem determination. Problem-determination output is used to determine the system resource responsible for the problem and the general area where

problem diagnosis will begin. If diagnostic data was gathered automatically with problem-determination data, it is used to assist in problem diagnosis. If additional data is needed, it is gathered and analyzed manually or with the help of problem management services. Some problems may require several iterations of this process, each time gaining more data and eliminating more components from the set of possible causes.

Overview

Problem diagnosis is composed of the following steps:

Collection of Diagnostic Data

The collection step of problem diagnosis gathers information necessary to support the analysis step of problem diagnosis. Problem-diagnosis data is typically unique to an implementation and is defined in implementation documentation. The transport of that data, if it is available, is defined by management services. The collection step may be performed by any of the following:

- Local management services or CPMS (In these cases the diagnostic data or problem-resolution data is transported in an Alert.)
- Personnel responsible for problem diagnosis using the following:
  - Serviceability aids and tools available at the node
  - Resource control commands Query Resource Data and Test Resource if supported at the node
  - Execute Command to transport requests for functions available at the node that are not otherwise within the scope of SNA/Management Services.

Analysis of Diagnostic Data

The analysis step of problem diagnosis processes diagnostic data to determine the precise cause of a problem and the precise action required for its resolution. The analysis step may be performed by any of the following:

- Local management services or CPMS (In these cases the diagnostic data or problem-resolution data is transported in an Alert.)
- Personnel responsible for problem diagnosis using any of the following:
  - SNA problem management services at the control point
  - Serviceability aids and tools available at the node.
  - Manual analysis of diagnostic data either at the node or at the control point
  - Use of the probable cause data reported by Analyze Status if it is supported at the node.

The output of this step is referred to as problem-resolution data.

Reporting and Presentation of Problem-Resolution Data

The reporting and presentation step of problem diagnosis reports the data required for problem resolution as follows:

- If personnel responsible for problem diagnosis performed the collection and analysis, it sends the problem-resolution data to them.
- If local management services performed collection and analysis, it sends the data to PUMS, which in turn sends the data to CPMS via an Alert.
- If CPMS performed collection and analysis, it sends the data to the Alert-processing function.

Processing Requests for Diagnostic Data

The request step of problem diagnosis provides a mechanism for the operator to allow diagnostic data to be requested for those cases where collection, analysis, and reporting are not performed automatically by a system process and must be controlled by personnel responsible for problem diagnosis.

Problem Determination and Problem Diagnosis via Alerting

Functions Provided

The PUMS component of each PU within a network provides an unsolicited notification when a problem is detected with any system resource that it manages or uses. That notification is an Alert. For details on the types of problems to be reported, refer to "Problem Detection" on page 3-6. The unsolicited notification is reported to the control point for the failing system resource, and provides a method for transporting data related to the problem.

Alerts are processed (filtered) at the control point to determine if they should be stored or presented to a network operator.

The advantages of using unsolicited notifications are as follows:

- They facilitate the managing of a network from a central location.
- They provide immediate notification to the network operator of the existence of a problem and the action to be taken. Alerts allow the network operator to be aware of the problem before the end user complains about the loss of availability, thus aiding responsiveness to and communication with the end user.
- They provide for the transport of data captured at the time of failure rather than having to depend on later re-creation techniques that may not be effective.
- They provide for automating problem determination and problem diagnosis, thus decreasing the time and skill level required to resolve problems.

Relationship of Errors, Problems, and Alerts

The relationship of errors, problems, and unsolicited notifications must be understood before the conditions under which Alerts are sent can be understood. Before continuing, the reader should be familiar with the following terms: *Alert*, *Alert condition*, *error*, *error condition*, *problem*, and *impending problem*. Refer to the glossary at the back of this manual for precise definitions.

We can see from the definitions of the previously mentioned terms that error conditions become problems only when they affect availability, and become impending problems only when they threaten to affect availability. Also, problems or impending problems are reported by unsolicited notifications only when they require action by the network operator.

The term "impending problem" covers two different senses in which a condition may threaten to affect availability:

- If a quantity, such as temperature or utilization, will affect availability if it exceeds a particular threshold, then it threatens to affect availability as it approaches this threshold.
- If a piece of redundant hardware or software fails, then it threatens to affect availability by reducing the level of redundancy in place to handle the next failure. This is true regardless of whether the initial failure occurrs on the then-active component, forcing a switchover to a backup, or on a backup component not currently in use.

The Alert is sent to communicate the existence of a problem or impending problem for which some or all of the process of problem analysis has been completed, taking into consideration all of the available problem data, by the node detecting the problem.

Problems can be detected only in those cases where a loss of availability can be detected. The following methods may be used to detect a loss of availability.

- Availability can be measured in those cases where it is possible to predict the performance that can be expected. If this can be done, it is possible to detect a performance level that deviates from the norm by an amount that for all practical purposes constitutes a loss of availability to the end-user and thus can be defined as a problem. This procedure is referred to as *performance analysis*.
- Rather than directly detecting a loss of availability, a procedure referred to as *error-rate analysis* can be used to infer loss of availability.

  Error conditions are divided into two categories, irrecoverable and recoverable, which vary in their effects on availability.

  - *Irrecoverable error condition*
    - The occurrence of *n* consecutive errors of certain types constitutes an irrecoverable error condition. The number *n* varies depending on the type of error, implementation, or technology, and defines a threshold beyond which subsequent retries of the operation that

prompted the initial error have little or no chance of resulting in a successful operation. Other thresholding techniques may also be used to determine an irrecoverable error condition. The technique used depends on the product or the technology involved.

— For errors of certain other types, a single occurrence constitutes an irrecoverable error condition. The operation resulting in the error is not retried. An example of this type of error is an "off-line," "unavailable," or "data unsafe" condition indicated via status or sense information.

Unless specifically defined by the architecture, it is left to each implementation to determine what constitutes an irrecoverable error condition. Thresholds for these purposes may be set and/or modified by system definition parameters or by local or remote operators.

Most irrecoverable error conditions cause a loss of availability of a system resource and thus fall into the category of problems. If, however, the operation requested by the end user is completed without the end user's knowledge of a loss of availability, the irrecoverable error condition is not classified as a problem. This may be the case in devices that have redundant hardware logic or other alternate means of performing an operation. These cases fall into the category of impending problems since the error condition cannot be resolved and the affected component will not be available without external intervention.

Note that an error that is unrecoverable at one level within a node may be recoverable at a different level within the same node. In this section the term "irrecoverable error" indicates as error that is not recoverable at *any* level within the node experiencing it.

— *Recoverable error condition*

Recoverable error conditions can be categorized as problems or impending problems when they have been judged to cause, or to have the potential for causing, a loss of availability to the end user. The following criteria may be used to make this judgment:

— Availability may be affected when the accumulated number of recoverable error conditions equals some preset threshold. The threshold may be described as x failures out of y attempts, or x failures in t time. It may also be described as x consecutive failures followed by a good operation, as long as x is less than the threshold used to determine an irrecoverable error condition.

— Availability may be affected by a single occurrence of an error such as exceeding a buffer threshold, storage usage threshold, or queue-length threshold.

Unless specifically defined by the architecture, it is left to each implementation to determine what constitutes a loss of availability; such thresholds used to make this determination are dependent upon the error type and the device and are considered to be implementation unique. They may be set and/or modified by system definition parameters or by local or remote operators.

If both performance analysis and error-rate analysis are used to indicate the existence of a problem, two indications may exist for the same problem. This may aid in problem determination since it may indicate that it was the error condition that caused the loss of availability detected also as abnormal performance.

Errors conditions that do not affect availability and thus are not reported as problems may be logged locally. This data will be available, if required, for problem diagnosis. Logging of error data is an implementation decision based on the service requirements. Another option for error conditions that do not immediately affect availability is to report them as impending problems. Typically, this type of reporting is used for cases where temperatures or voltages are out of specification but the device continues to function normally. Reporting as an impending problem can also be used for error rates that do not affect availability but are judged by an implementation to be beyond what is normally expected.

Variations in Alert Data

Because of the variety of problems that may be encountered and the various ways that they may manifest themselves, it is not always possible for products to generate Alerts that contain complete problem determination and problem diagnosis information. The network operator may be presented data that represents any of the following situations:

- A problem has been detected but analysis is not complete. The origin and general cause of the problem have not been determined. Some data may have been gathered.

  Action Required by Network Operator

  The network operator is responsible for any additional collection of problem data that may be required, and for analysis to determine the general cause and the resource that is the origin of the problem. Collection may consist of: (1) retrieving data that was collected at the time of the failure but was not sent with the Alert; (2) running diagnostics to attempt to re-create the failure and collect data about it. User procedures are then employed to determine the organization responsible for problem diagnosis for further handling of the problem.

- A problem has been detected, analysis has been completed, and the origin and general cause of the problem have been determined.

  Action Required by Network Operator

  The network operator is responsible for employing user procedures to determine the organization responsible for problem diagnosis and further handling of the problem.

- A problem has been detected, analysis has been completed, and the origin and general cause of the problem have been determined. Problem diagnosis data has been collected but analysis of that data is not complete. The action required for problem resolution is not known.

  Action Required by Network Operator

The network operator is responsible for employing user procedures to determine the organization responsible for problem diagnosis. The problem and the collected problem diagnosis data are turned over to that organization.

- A problem has been detected. Analysis has been completed and the origin and general cause of the problem have been determined. Problem diagnosis data has been collected and analysis of that data is complete. The action required for problem resolution is known.

  Action Required by Network Operator

  The network operator is responsible for using user procedures to determine the organization responsible for problem resolution. The problem and the collected problem resolution data are turned over to that organization.

Figure 3-4 on page 3-16 shows the minimum function supplied by the Alert (the transport of problem data or problem-determination data). Figure 3-5 on page 3-17 shows the maximum function supplied by the Alert (the transport of problem determination and diagnostic or problem-resolution data). The differences between the two figures have been underscored in Figure 3-5.

Note that in Figure 3-4, if complete problem-determination data is not presented to the network operator, the operator must use manual procedures to perform collection and analysis until problem determination is complete and the problem may be given to the personnel responsible for problem diagnosis. In Figure 3-5, if problem diagnosis has been completed and complete problem-resolution data is presented to the network operator, the network operator will contact the personnel responsible for problem resolution. If problem diagnosis is not complete, the network operator will contact the personnel responsible for problem diagnosis to use manual procedures to perform collection and analysis until problem diagnosis is complete and the problem may be given to the personnel responsible for problem resolution.

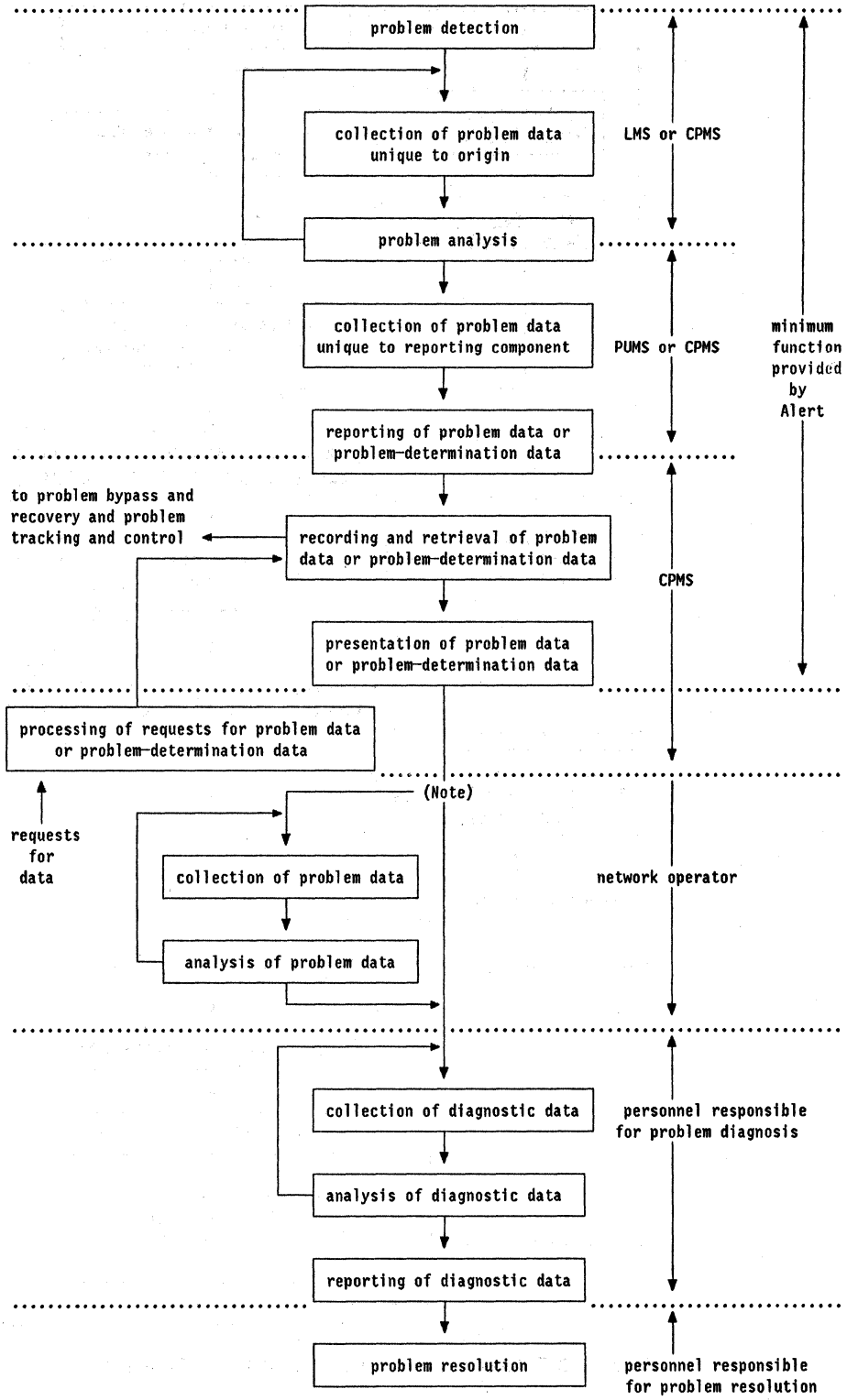


Figure 3-4. Minimum Function Supplied by the Alert.

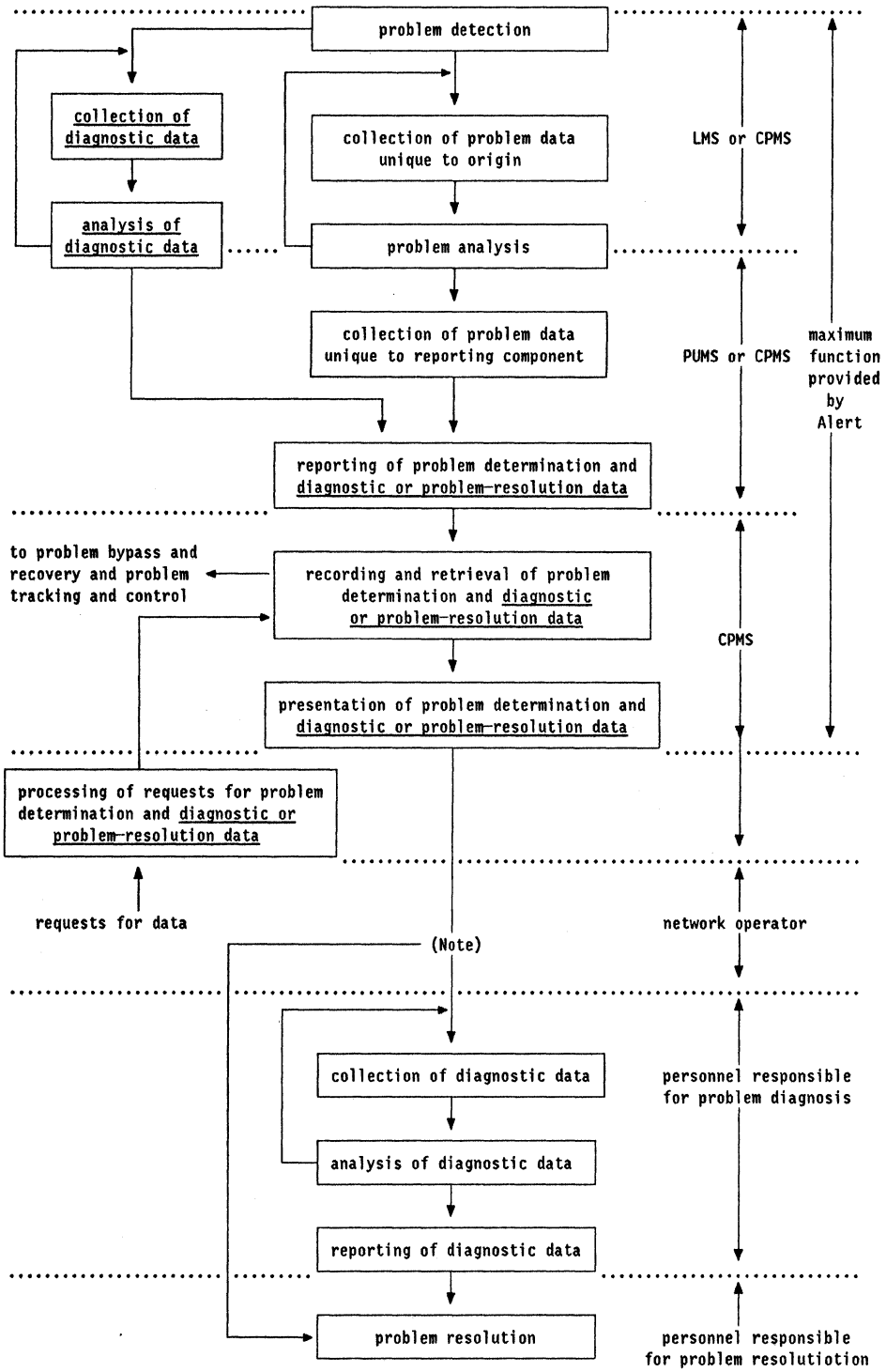


Figure 3-5. Maximum Function Supplied by the Alert.

Implementation

Alert support is provided in a PU by the "EP_ALERT Function Set" on page 10-3. This is a base function implemented by all nodes.

Format Usage

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Alert Major Vector (X'0000')
Flow: From PU to SSCP (Normal)

An Alert may be initiated when any nodal component detects a condition indicating the loss or impending loss of availability of a system resource to an end user. It is thus a flow of the type described in "Unsolicited Flows" on page 2-8. Information to be included in an Alert (X'0000') major vector is passed from the initiating LMS to PUMS. The Alert major vector contains the type of the Alert, a general classification and cause of the condition being reported, the identity of the resource to which the Alert applies, and additional information to assist in problem determination. PUMS constructs an NMVT containing this major vector and sends it to CPMS.

When CPMS receives the NMVT, it optionally logs the data, and passes it to the network operator. Refer to Figure 3-6 for an example flow.

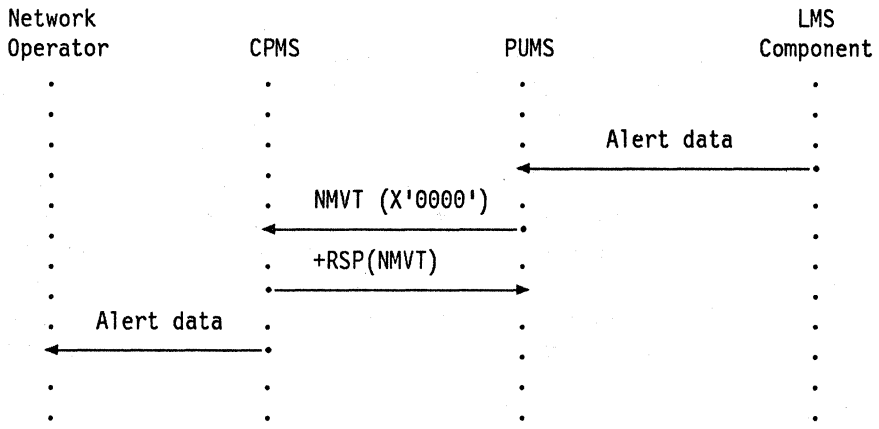


Figure 3-6. Example Flow Showing an Alert

Figure 3-7 on page 3-19 illustrates a special type of Alert, the *delayed* Alert. If the condition causing the Alert interrupts communication between PUMS and CPMS, the Alert cannot be sent immediately. In this case, PUMS stores the Alert until communication with CPMS has been re-established. It then sends the Alert to CPMS.

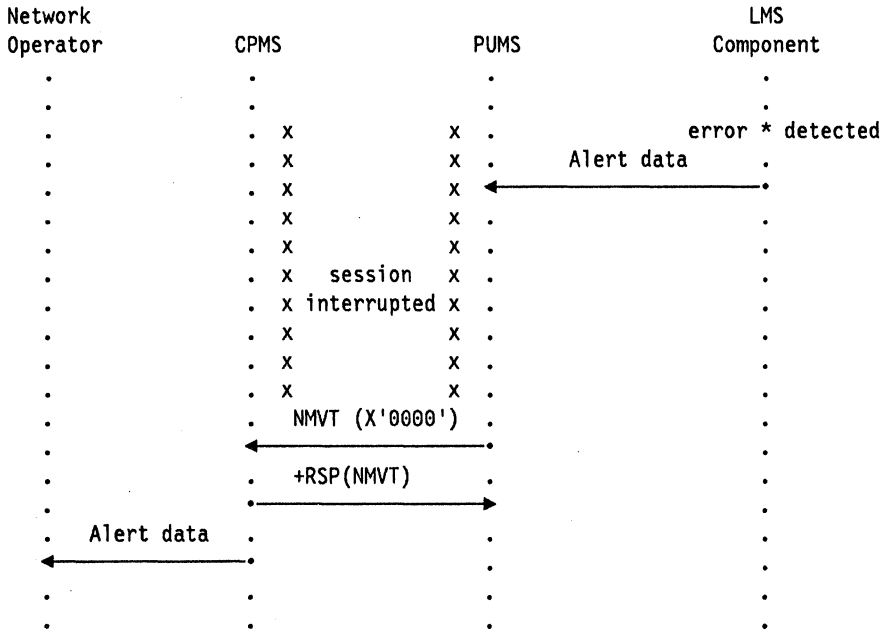


Figure 3-7. Example Flow Showing Delayed Alert. The Alert is reporting a condition that caused it to be delayed, in this case a condition that interrupted the SSCP-PU session.

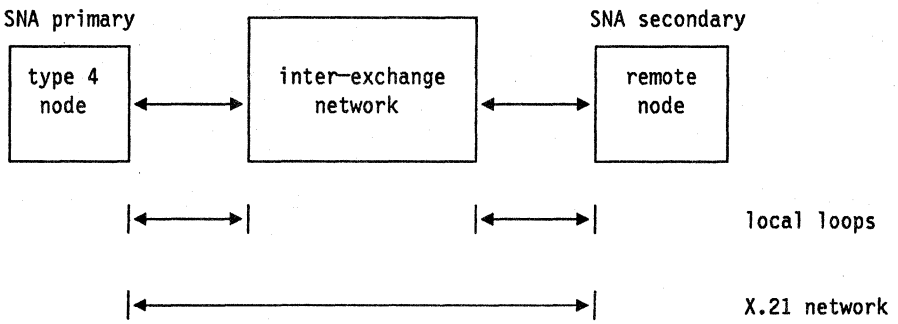
Related to delayed Alerts are *held* Alerts. Like a delayed Alert, a held Alert cannot be sent immediately because of an interruption in communication between PUMS and CPMS. The held Alert, however, does not report the condition that caused the interruption; it instead reports some (typically) unrelated condition, that just happened to be detected when communication was interrupted. The key difference between held and delayed Alerts lies in the fact that if a delayed Alert successfully reaches CPMS, then it follows that the problem it reports has been either resolved or bypassed, at least temporarily. Otherwise, the delayed Alert could not have reached CPMS. With a held Alert, no such inference is possible: the problem it reports may have been resolved or bypassed, but it is equally possible that it is still impacting availability.

Link Services

Link services assists in problem determination and problem diagnosis for problems associated with communication links.

Managing Links

Links Traversing X.21 Networks



Note: The remote node can be either a type 2 node or a boundary-function-attached type 2.1 node

Figure 3-8. Boundary Function Attachment Over an X.21 Network

Functions Provided:

Figure 3-8 shows a type 4 node communicating with either a type 2 node or a boundary-function-attached type 2.1 node over a link established by means of an X.21 Network. For an introduction to X.21 networks see *CCITT Recommendation, X.21*, 1984.

The management services support unique to X.21 link connections is limited to one function: issuing Alerts for such conditions as unsuccessful connection attempts and network timeouts. The type 2 node or the boundary-function-attached type 2.1 node issues a delayed Alert for an unsuccessful short-hold mode *reconnection*. The delayed Alert is sent only if the type 4 node with which the sender establishes a connection is the same one to which the reconnection attempt was directed; if a different type 4 node is reached, no Alert is sent.

No Alert is issued by a type 2 or boundary-function-attached type 2.1 node for an unsuccessful initial connection, since at this point the sending node would be unknown to the network, and thus the Alert would serve no purpose. (For a description of the format that supports the establishment of a short-hold mode connection, see the Format 1 continuation of XID, in *Systems Network Architecture Formats*, GA27-3136.)

Implementation:

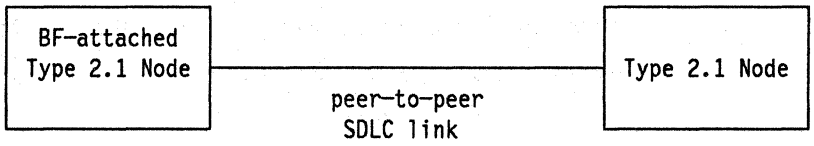
Alert support in the boundary-function-attached type 2.1 node and the type 2 node includes Alerts issued for the X.21 network. See "EP_ALERT Optional Subset 9 (X.21 Alert)" on page 10-22.

Format Usage:

The only format involved in the management of the links traversing X.21 net-

works is the Alert. See "X.21 and X.21 Short Hold Mode Alerts" on page A-69 for a list of the Alerts defined for links traversing X.21 networks.

SDLC Links



Note: BF-attached denotes boundary-function-attached

Figure 3-9. Peer-to-Peer SDLC Link Between Type 2.1 Nodes. The boundary-function-attached type 2.1 node may contain either the primary or the secondary station for the SDLC link.

Functions Provided:

Figure 3-9 shows a peer-to-peer SDLC link between a boundary-function-attached type 2.1 node and a remote type 2.1 node. In this configuration, the boundary-function-attached type 2.1 node sends Alerts to its SSCP for the logical link with its peer type 2.1 node. The boundary-function-attached type 2.1 node may contain either the primary or the secondary station on the SDLC connection, and will send different Alerts accordingly.

Implementation:

Alert support in the boundary-function-attached type 2.1 node includes Alerts issued for the SDLC logical link with its peer type 2.1 node. See "EP_ALERT Optional Subset 8 (SDLC/LAN LLC Alert)" on page 10-22.

Format Usage:

The only format involved in the management of the SDLC logical link between a boundary-function-attached type 2.1 node and its peer type 2.1 node is the Alert. See "SDLC Alerts" on page A-55 for a list of the Alerts defined for SDLC logical links.

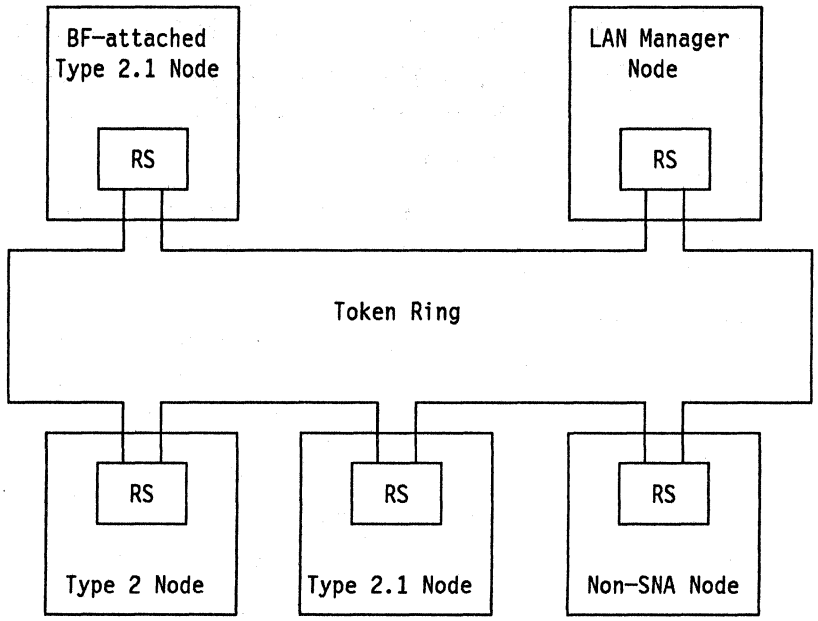
Links Traversing Local Area Networks

The next four sections discuss four types of management for Local Area Networks (LANs):

- Medium-Access Control- (MAC-) level management for token-ring LANs
- MAC-level management for Carrier Sense Multiple Access with Collision Detection (CSMA/CD) LANs
- MAC-level management for bridged LANs; the LAN bridges may connect two token rings, two CSMA/CD busses, or one ring and one bus
- Logical Link Control- (LLC-) level management for logical links traversing any of the three types of LANs listed above

Note that management of links traversing local area networks involves management at both the MAC and LLC levels. Thus a node responsible for managing such a link must implement the functions described in "LLC-Level Management for Links Traversing Local Area Networks" on page 3-30 in addition to those described for the appropriate MACs.

MAC-Level Management for Token-Ring LANs:



- RS denotes ring station
- BF-attached denotes boundary-function-attached

Figure 3-10. Nodes Communicating Over a Token-Ring LAN

Functions Provided:

Figure 3-10 shows an example of a token-ring LAN and its attached nodes. Five nodes attached to the LAN are shown, each with a different role in the management process:

- Boundary-function-attached Type 2.1 node
  - Issues Alerts concerning the local LAN adapter (hardware check, programming error, and removal from the ring), wire faults, and ring inoperative conditions. (see "Problem Determination and Problem Diagnosis via Alerting" on page 3-11 for details on Alerts)

    Note: The Alerts reporting the ring inoperative conditions are not sent by this node if a LAN manager is managing the ring to which it is attached. Knowledge of whether a LAN manager is present on a ring must be defined locally at each Alert sender on the ring; there is no mechanism for acquiring this knowledge dynamically.
  - Participates in the MAC-level management functions required for management and operation of the ring. (See *Token-Ring Network Architecture Reference*, SC30-3374, for details on the operation of token-ring LANs)

- Type 2 node:
  - Participates in the MAC-level management functions required for management and operation of the ring
- Type 2.1 node (not boundary-function-attached):
  - Participates in the MAC-level management functions required for management and operation of the ring
- Non-SNA node:
  - Participates in the MAC-level management functions required for management and operation of the ring
- LAN Manager
  - Issues Alerts concerning the local LAN adapter (hardware check, programming error, and removal from the ring), wire faults, remote LAN adapter removal from the ring, ring inoperative conditions, excessive soft errors, and program errors in the LAN manager (see "Problem Determination and Problem Diagnosis via Alerting" on page 3-11 for details on Alerts)
  - The LAN Alert Transport is contained within the LAN Manager and provides an "Alert pass-through" for sending Alerts to the focal point. Devices and software on the LAN that experience error conditions can build Alert messages and send them over the LAN to the LAN Manager. The LAN Manager receives the Alert message and sends an Alert to the focal point.
  - Participates in the MAC-level management functions required for management and operation of the ring

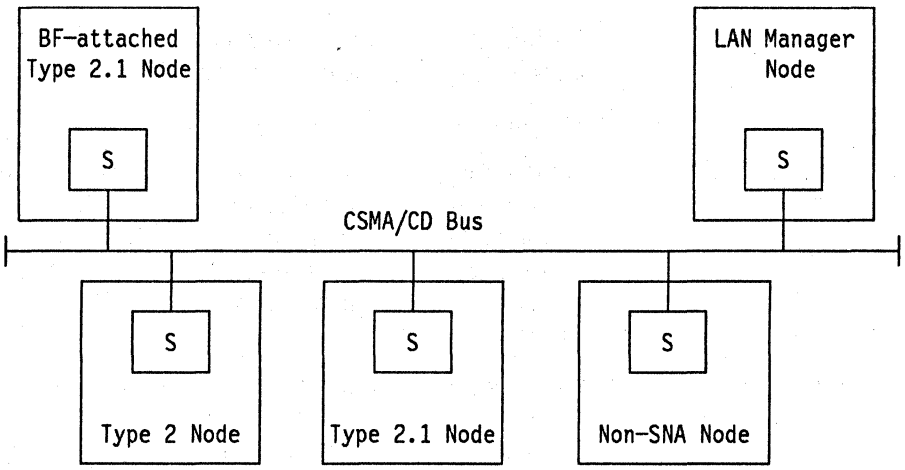
Implementation:

Alert support in the LAN Manager and in the type 2.1 nodes includes Alerts issued for the token-ring LAN link connection. See "EP_ALERT Optional Subset 7 (LAN Alert)" on page 10-21.

Format Usage:

The only format involved in the MAC-level management of a token-ring LAN. is the NMVT Alert. See "Token-Ring LAN Alerts" on page A-1 for a list of the Alerts defined for a node providing MAC-level management for a token-ring LAN.

MAC-Level Management for CSMA/CD LANs:



- S denotes station
- BF-attached denotes boundary-function-attached

Figure 3-11. Nodes Communicating Over a CSMA/CD LAN

Functions Provided:

Figure 3-11 shows an example of a CSMA/CD LAN and its attached nodes. Five nodes attached to the LAN are shown, each with a different role in the management process:

- Boundary-function-attached Type 2.1 node
  - Issues Alerts concerning the local LAN adapter (hardware check, programming error, removal from the bus), lack of signal at the adapter, and bus-inoperative conditions. (see "Problem Determination and Problem Diagnosis via Alerting" on page 3-11 for details on Alerts)

    Note: The Alerts reporting the bus-inoperative conditions are not sent by this node if a LAN manager is managing the bus to which it is attached. Knowledge of whether a LAN manager is present on a bus must be defined locally at each Alert sender on the bus; there is no mechanism for acquiring this knowledge dynamically.
  - Participates in the MAC-level management functions required for management and operation of the bus
- Type 2 node:
  - Participates in the MAC-level management functions required for management and operation of the bus
- Type 2.1 node (not boundary-function-attached):

  - Participates in the MAC-level management functions required for management and operation of the bus

- Non-SNA node:
  - Participates in the MAC-level management functions required for management and operation of the bus
- LAN Manager
  - Issues Alerts concerning the local LAN adapter (hardware check, programming error, removal from the bus), lack of signal at the adapter, bus-inoperative conditions, unauthorized adapter insertion attempts, and removal of remote adapters from the bus. (see "Problem Determination and Problem Diagnosis via Alerting" on page 3-11 for details on Alerts)
  - The LAN Alert Transport is contained within the LAN Manager and provides an "Alert pass-through" for sending Alerts to the focal point. Devices and software on the LAN that experience error conditions can build Alert messages and send them over the LAN to the LAN Manager. The LAN Manager receives the Alert message and sends an Alert to the focal point.
  - Participates in the MAC-level management functions required for management and operation of the bus

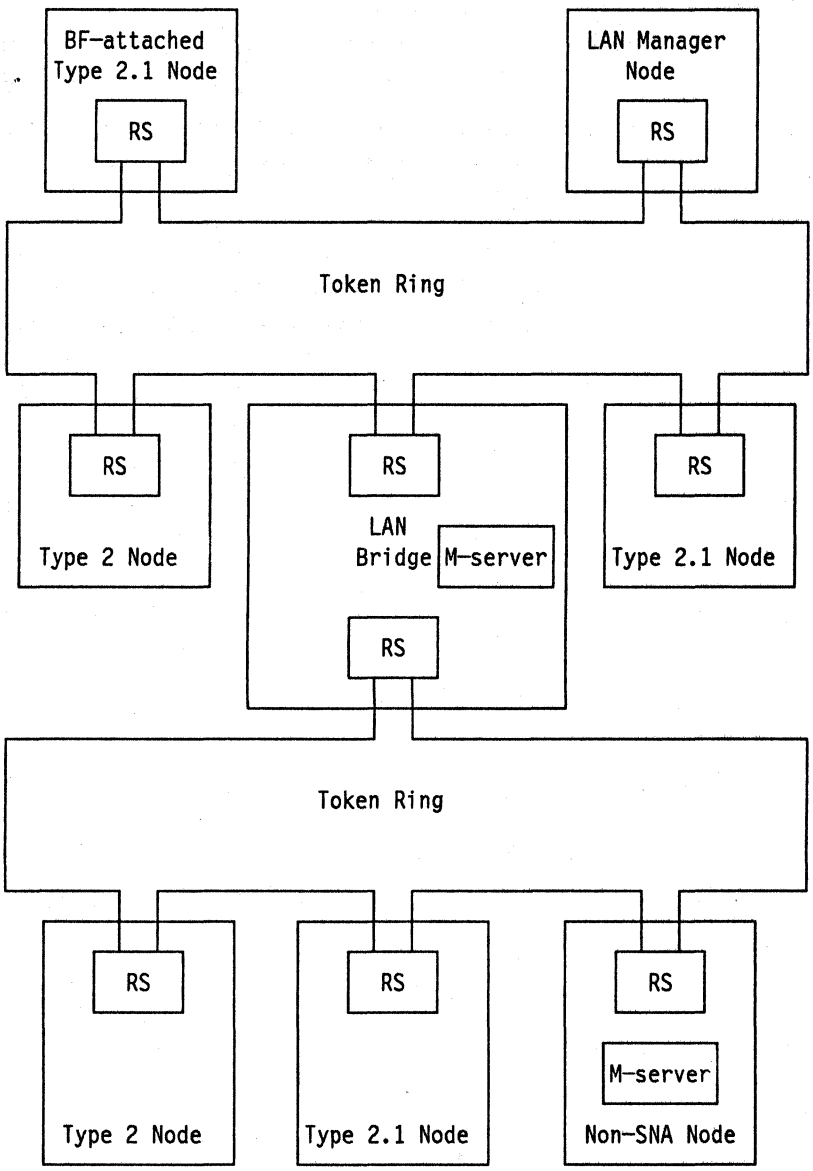
Implementation:

Alert support in the LAN Manager and in the type 2.1 nodes includes Alerts issued for the CSMA/CD LAN link connection. See "EP_ALERT Optional Subset 7 (LAN Alert)" on page 10-21.

Format Usage:

The only format involved in the MAC-level management of a CSMA/CD LAN is the NMVT Alert. See "CSMA/CD LAN Alerts" on page A-13 for a list of the Alerts defined for a node providing MAC-level management for a CSMA/CD LAN.

MAC-Level Management for Bridged LANs:



- RS denotes ring station
- M-server denotes LAN management server
- BF-attached denotes boundary-function-attached

Figure 3-12. Nodes Communicating Over a Bridged Token-Ring LAN. This figure shows a bridge between two token rings; a bridge may also be between two CSMA/CD busses, or between a token ring and a CSMA/CD bus.

Functions Provided:

Figure 3-12 shows an example of a bridged LAN and its attached nodes. Five types of nodes, plus the bridge, are shown attached to the LAN, but the only

ones with specific responsibilities for managing the bridged LAN are the LAN Manager and those nodes attached to the remote ring (or bus) that contain a LAN management server:

- Node that is attached to the remote ring or bus and contains a LAN management server; this node may, but need not be, the bridge itself
  - Assembles and analyzes MAC-level from the other nodes attached to its ring or bus, and forwards the results of this analysis to the LAN manager
- LAN Manager
  - Issues Alerts concerning the bridges and remote rings or busses it is responsible for, as well as its logical connections to the LAN management servers that provide it data about these bridges, rings, and busses. (see "Problem Determination and Problem Diagnosis via Alerting" on page 3-11 for details on Alerts)
  - The LAN Alert Transport is contained within the LAN Manager and provides an "Alert pass-through" for sending Alerts to the focal point. Devices and software on the LAN that experience error conditions can build Alert messages and send them over the LAN to the LAN Manager. The LAN Manager receives the Alert message and sends an Alert to the focal point.
  - Participates in the MAC-level management functions required for management and operation of the bridged LAN

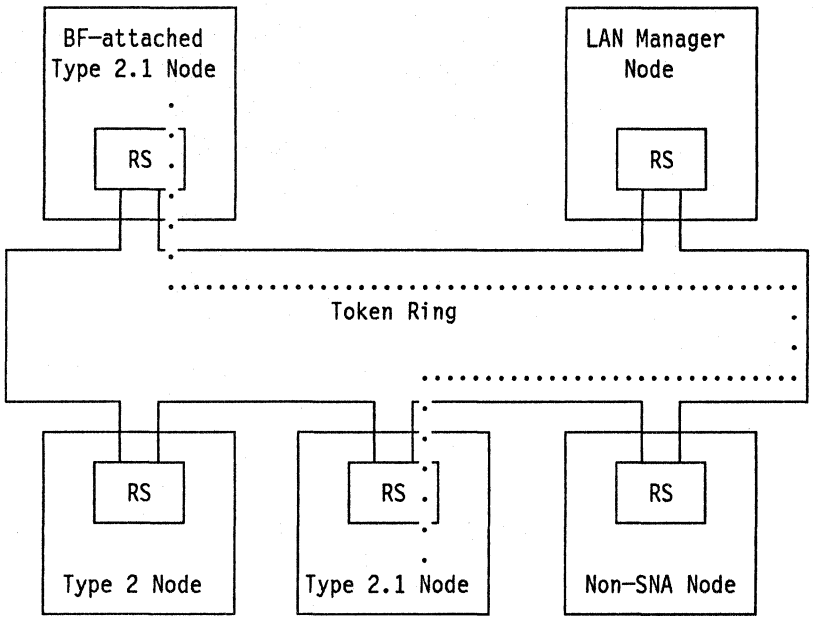
Implementation:

Alert support in the LAN Manager includes Alerts issued for the bridged LAN.

Format Usage:

The only format involved in the MAC-level management of a bridged LAN is the Alert. See "Bridged LAN Alerts" on page A-21 for a list of the Alerts defined for a LAN manager providing MAC-level management for a bridged LAN.

LLC-Level Management for Links Traversing Local Area Networks



- RS denotes ring station
- BF-attached denotes boundary-function-attached
- ..... denotes a peer-to-peer (type 2.1 to type 2.1) logical link connection

Figure 3-13. Logical Links Traversing a Token-Ring LAN. While the figure shows a token-ring LAN, CSMA/CD and bridged LANs also support such logical links. In a bridged LAN, a logical link may traverse a bridge.

Functions Provided:

Figure 3-13 shows an example of a local area network and a peer-to-peer logical link, between two type 2.1 nodes, that traverses it. The boundary-function-attached type 2.1 node is responsible for issuing Alerts on the peer-to-peer logical link.

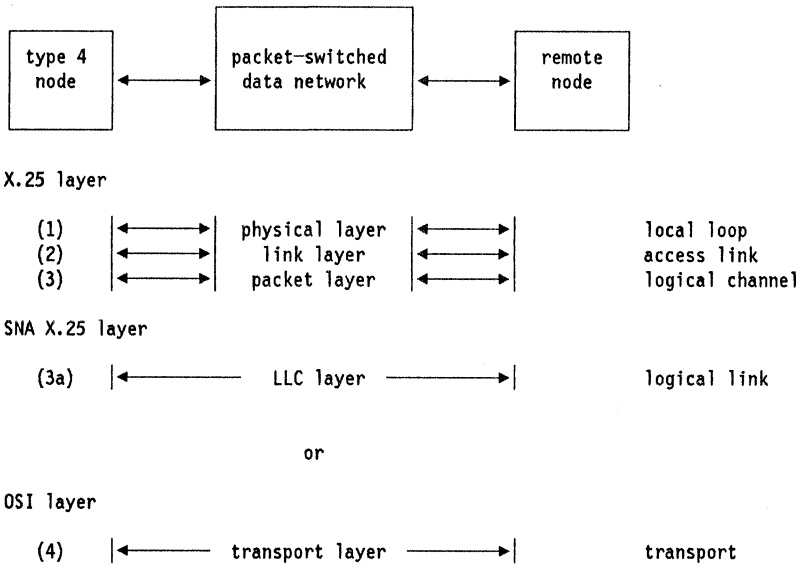
Implementation:

Alert support in the boundary-function-attached type 2.1 node includes Alerts issued for logical link connections traversing a LAN. See "EP_ALERT Optional Subset 8 (SDLC/LAN LLC Alert)" on page 10-22.

Format Usage:

The only format involved in the management of LAN logical link connections is the NMVT Alert. See "LAN LLC Alerts" on page A-41 for a list of the Alerts defined for a node providing LLC-level management for links that traverse LANs.

Links Traversing X.25 Packet-Switched Data Networks



Note: The remote node can be either a type 2 node or a boundary-function-attached type 2.1 node

Figure 3-14. Boundary Function Attachment Over an X.25 Network

Functions Provided:

Figure 3-14 shows a type 4 node communicating with a type 2 or a type 2.1 boundary-function-attached remote node over a link established by means of an X.25 Packet-Switched Data Network. For an introduction to X.25 networks see *The X.25 1984 Interface for Attaching SNA Nodes to Packet-Switched Data Networks General Information Manual*, IBM Publication GA27-3761. Management services support is provided for X.25 connections by providing for the issuance of Alerts.

Implementation:

The Alert support required for management services of X.25 link connections may be found in "EP_ALERT Optional Subset 11 (X.25 Alert)" on page 10-24.

Format Usage:

The only format involved in the management of X.25 link connections is the Alert. See "Alerts for X.25 Link Connections" on page A-88 for a list of the Alerts defined for a node that implements the X.25 protocols. The flow is of the type described in "Unsolicited Flows" on page 2-8.

Chapter 4. Performance and Accounting Management

Performance and accounting management is the process of quantifying, measuring, reporting, and controlling the responsiveness, availability, utilization, and costs of a data processing or information system. Response-time monitoring is the only element of performance and accounting management for which SNA management services have been provided.

Response-Time Monitoring (RTM)

Response-time monitoring (RTM) provides the capability to quantify, measure, and report end-user response times for dependent LUs. Refer to Figure 4-1 for an overview of the steps required for RTM and the components in which the steps are implemented.

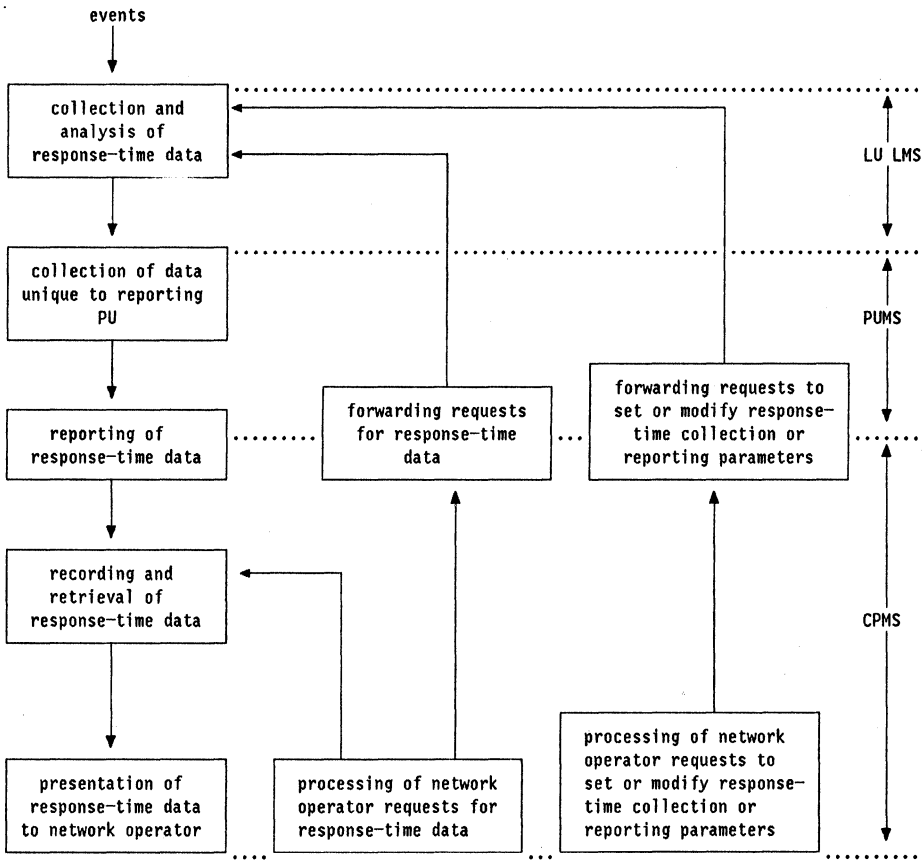


Figure 4-1. Overview and Placement of Response-Time Monitoring Steps

RTM provides for measurement of response time at the workstation, allows correlation of transaction timing with application usage, allows network owners to define how response times are to be measured, and allows the network operator to have real-time access to the response-time data.

RTM is performed automatically, although a mechanism may exist for a human operator to change the collection or reporting parameters, or to request response-time data.

RTM is only supported for dependent LUs.

Functions Provided

Collection and Analysis of Response-Time Data

The collection and analysis function of RTM provides the following:

- Measurement and collection of end-user response times at the workstation or controller according to definitions specified by the network operator (See "Processing and Forwarding of Network Operator Requests" on page 4-5.)
- Collection of data that will identify any potential inaccuracy of the response-time data, e.g., counter overflow or the activation of a new session before the data from the previous session could be transmitted
- Collection of data that will allow the correlation of response times with application usage, e.g., identification of primary and secondary LU
- Measurement of the total time period over which the data was collected.

Reporting of Response-Time Data

The reporting function of RTM provides for reporting of response-time data to the RTM focal point under the following conditions:

- When solicited by the network operator
- Unsolicited on session deactivation, if specified by a network operator command
- Unsolicited on counter overflow, if specified by a network operator command.

Recording and Retrieval of Response-Time Data

The recording and retrieval function of RTM is implemented at the focal point and provides the following:

- Determines if the response-time data is to be logged
- Provides data base management (e.g., storage, retrieval) of the collected data.

Presentation of Response-Time Data

The presentation function of RTM is implemented at the focal point and provides the formatting and presentation of response-time data to allow the network operator to view the following data:

- A summary of the data for a specific LU over a user-defined time period
- Detailed data for a specific session for a single collection period
- Long-term trend for a specific LU.

Processing and Forwarding of Network Operator Requests

This function provides a protocol boundary for the network operator to request the following:

- Presentation of RTM data that is either requested from the RTM historical data base or from the collection and analysis function.
- The setting of the following parameters in the collection and analysis function:
  - The conditions under which unsolicited RTM data is reported

    The following options are available:
    - Report on session deactivation
    - Report on counter overflow.
  - The response-time unit of measure
  - The number and size of the ranges in which response times are to be reported

    For example, range 1 could be defined as all transactions with a response time of less than one second, range 2 as transactions falling within one to two seconds, range 3 as two to four seconds, and so forth. For each transaction, the appropriate range counter is updated. Up to four response-time boundaries, defining up to five ranges, may be set, and up to five response-time counters will be supported.
  - The definition of response times

    The following measurement options are available:
    - From the attention or action key depression to the arrival back at the LU of the first character that can alter the presentation space
    - From the attention or action key depression until the LU is ready to accept input from its end user
    - From the attention or action key depression to the receipt and processing back at the LU of Change Direction (CD) or End Bracket (EB)
    - From the attention or action key depression to the receipt of the last character of the last message received prior to the next attention or action key depression.

Implementation

Support of RTM is provided in a PU by "EP_RTM Function Set" on page 10-51. This is an optional function set implemented by type 2 nodes.

Format Usage

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Request Response Time Monitor MV (X'8080')
Flow: From SSCP to PU T2.0 (Normal)

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Response Time Monitor MV (X'0080')
Flow: From PU T2.0 to SSCP (Normal)

Setting of RTM Parameters

A network operator can request that RTM collection and reporting parameters be set to specified values for all LUs associated with a particular PU or for a specified LU. Data necessary for the request is passed to CPMS.

CPMS formats the request into an NMVT containing a Request RTM (X'8080') major vector and sends it to PUMS. The flow is of the type described in "Request Without Reply Flows" on page 2-9.

When PUMS receives the NMVT it does the following:

- If the request was for a single LU, PUMS sends a request to the specified LU. Refer to Figure 4-2 for an example flow.
- If the request applies to all LUs, PUMS determines the list of LUs configured at the node, and sends requests to each of them. The flow is similar to Refer to Figure 4-3 on page 4-7 for an example flow.

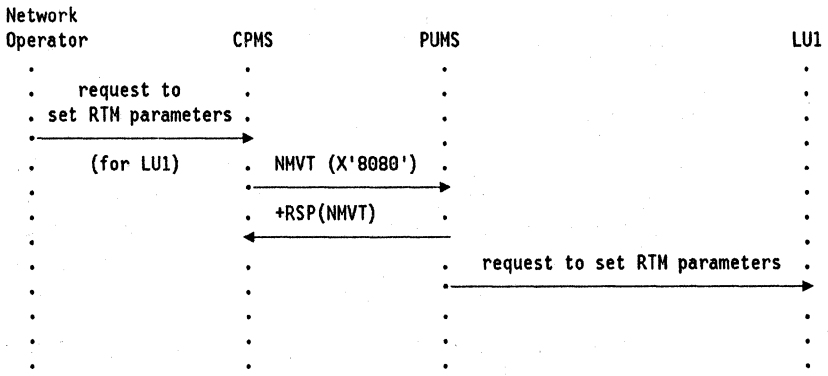


Figure 4-2. Example Flow Showing Request to Set RTM Parameters for a Single LU

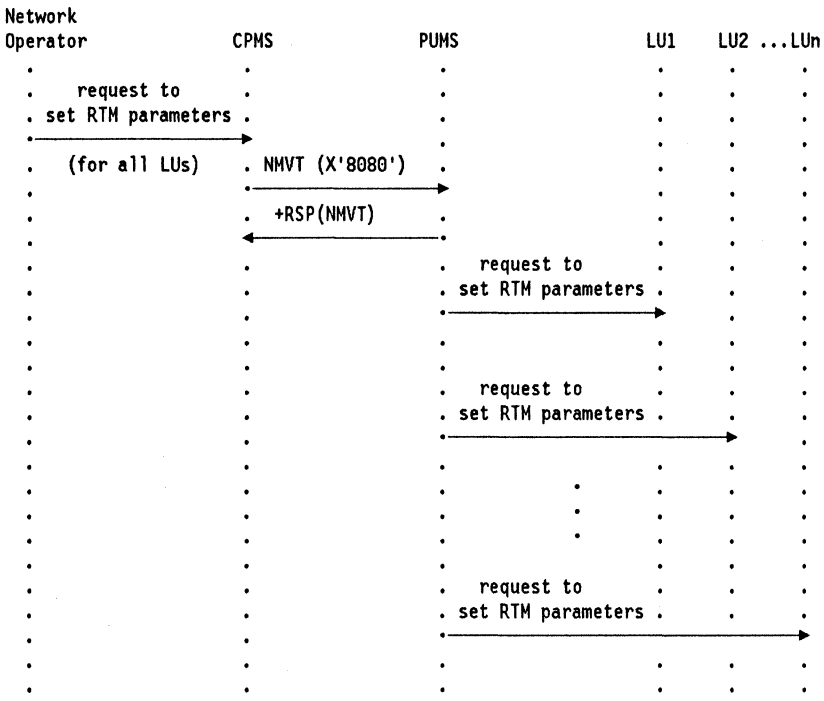


Figure 4-3. Example Flow Showing Request to Set RTM Parameters for All LUs

Requesting RTM Data and Status

A network operator can request RTM data from a PU for a single LU or for all LUs with accumulated data. Data necessary for the request is passed to CPMS.

CPMS formats the request into an NMVT containing a Request RTM (X'8080') major vector and sends it to PUMS. The flow is of the type described in "Request/Reply Flows for Non-Bulk Data" on page 2-10.

When PUMS receives the NMVT it does the following:

- If the request was for a single LU, PUMS sends a request to the specified LU. If the LU currently has RTM data, both accumulated data and status are returned; if it has no RTM data, or if it is inactive, only status information is returned. Refer to Figure 4-4 on page 4-8 for an example flow of the request for RTM data from a single LU.
- If the request applies to all LUs with accumulated data, PUMS determines the list of LUs configured at the node, and sends requests to each of them. Each LU responds to this request with its RTM status; if the LU currently has RTM data, it includes this data as well. For each LU that responds with both accumulated data and status, PUMS immediately builds an NMVT containing an RTM (X'0080') major vector and sends it to CPMS.

  PUMS discards responses from LUs that have no data, with the possible exception of the last one. An NMVT containing an RTM (X'0080') major vector with RTM status information may, if PUMS so elects, *always* be sent for the last LU, even if this LU returns no data to PUMS. By always sending a

reply for the last LU, PUMS is able to send replies for preceding LUs immediately, and thereby avoid the expenses associated with lookahead; otherwise it would have to hold each reply until it had determined whether or not that reply would be the last one for the request.

If *no* LU has accumulated RTM data, it sends a single reply containing RTM status information for its last configured LU. Refer to Figure 4-5 on page 4-9 and Figure 4-6 on page 4-10 for example flows.

When CPMS receives each NMVT, it optionally logs the data and passes it to the network operator.

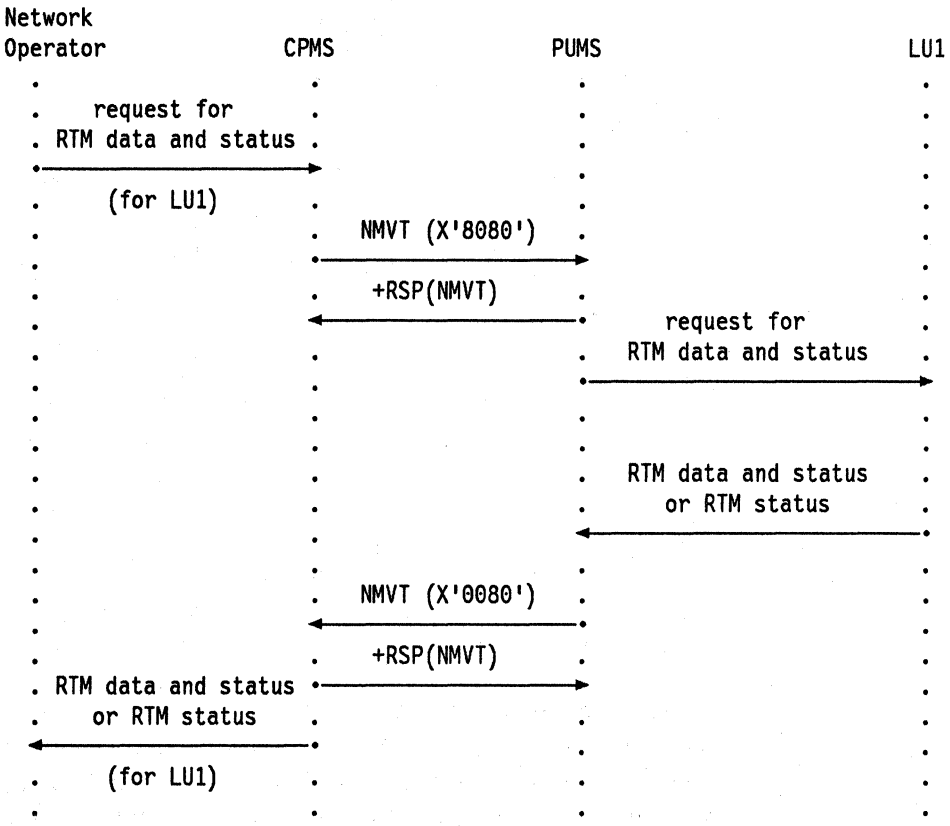
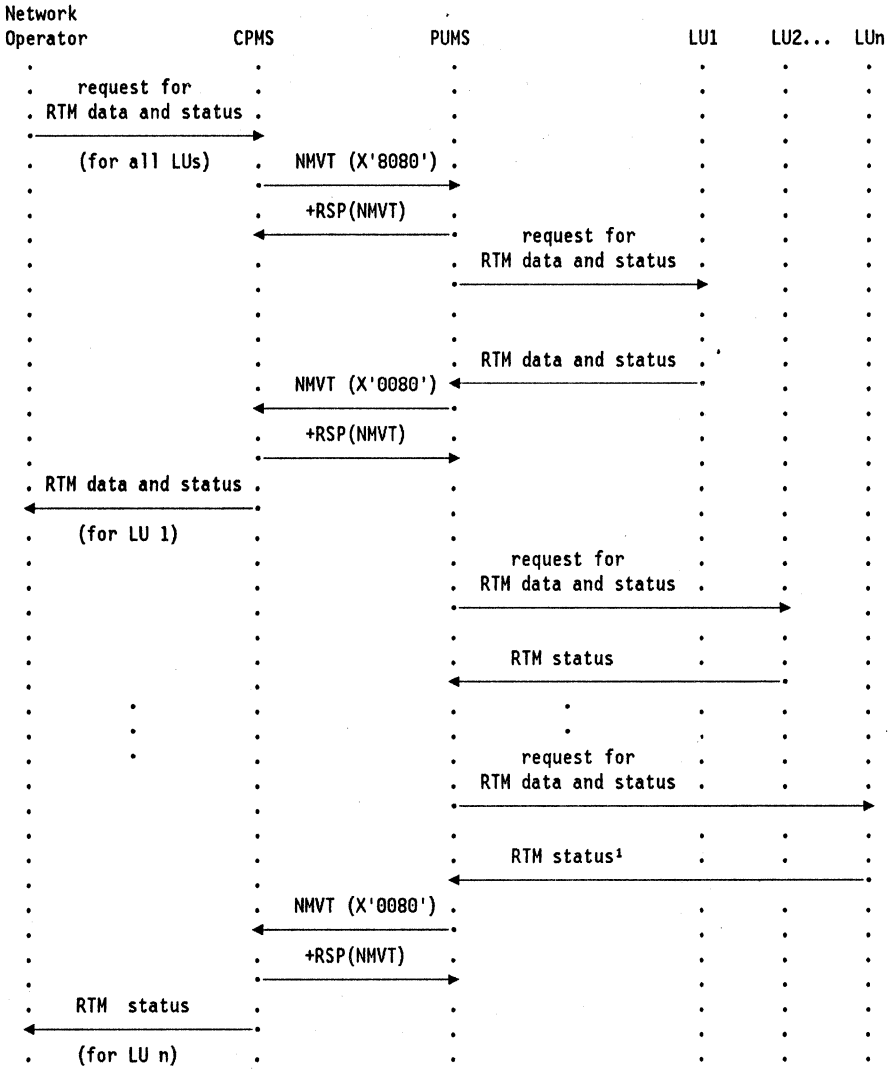


Figure 4-4. Example Flow Showing Request for RTM Data and Status for a Single LU, and the Resulting Reply



[1] The architecture allows a PU to reply with status only (no data) for its last defined LU if that LU has no accumulated RTM data. This prevents an implementation from having to buffer the data it has collected from all its LUs in order to determine whether the reply it is sending is the last one for the request.

Figure 4-5. Example Flow Showing Request for RTM Data and Status for All LUs, and the Resulting Replies

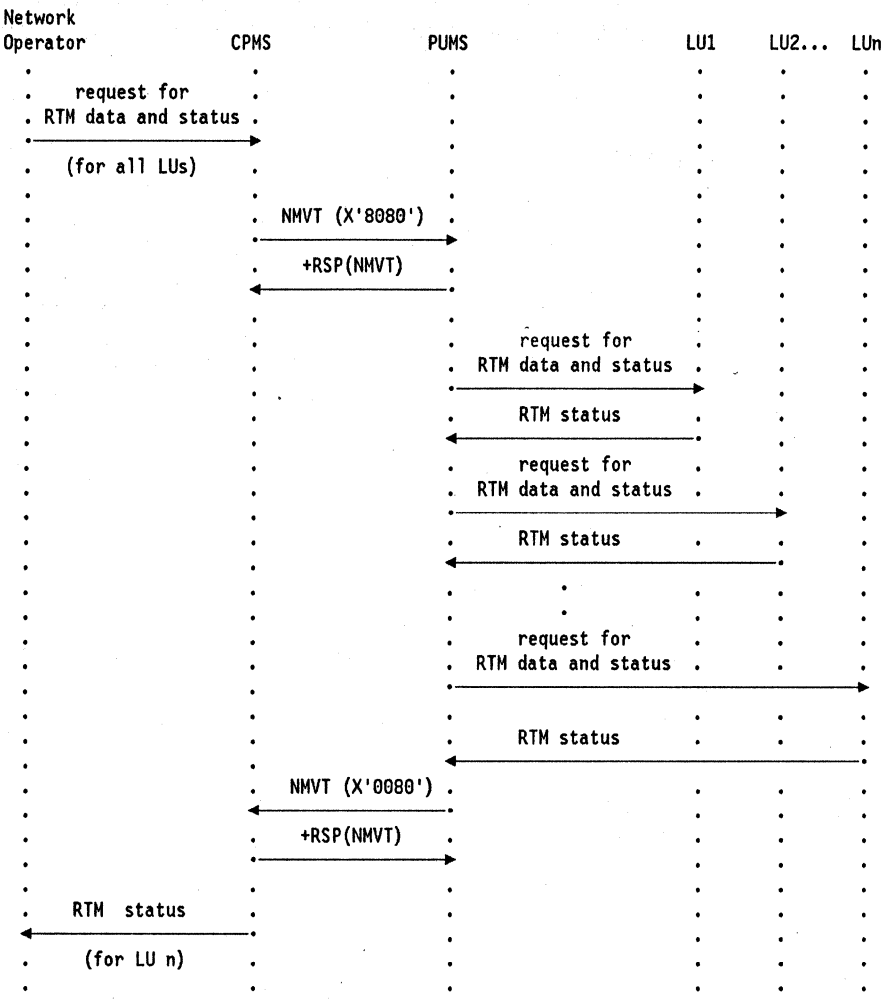


Figure 4-6. Example Flow Showing Request for RTM Data and Status for All LUs, When No LUs Have Accumulated Data. PUMS returns status only (no data) for its last defined LU.

Sending Unsolicited RTM Data and Status

RTM data and status can be sent unsolicited to a control point under the following conditions, if specified by the RTM reporting parameters:

- When session deactivation is detected
- When a response-time counter overflow is detected

Upon detection of one of the above conditions, the LU LMS sends RTM data and status to PUMS. PUMS formats the information into an NMVT containing an RTM (X'0080') major vector and sends it to CPMS. In the case of counter overflow, PUMS and the LU LMS may also be configured to send an Alert reporting the condition.

When CPMS receives the NMVT, it optionally logs the data and passes it to the network operator. The flow is of the type described in "Unsolicited Flows" on page 2-8. Refer to Figure 4-7 on page 4-11 for an example flow.

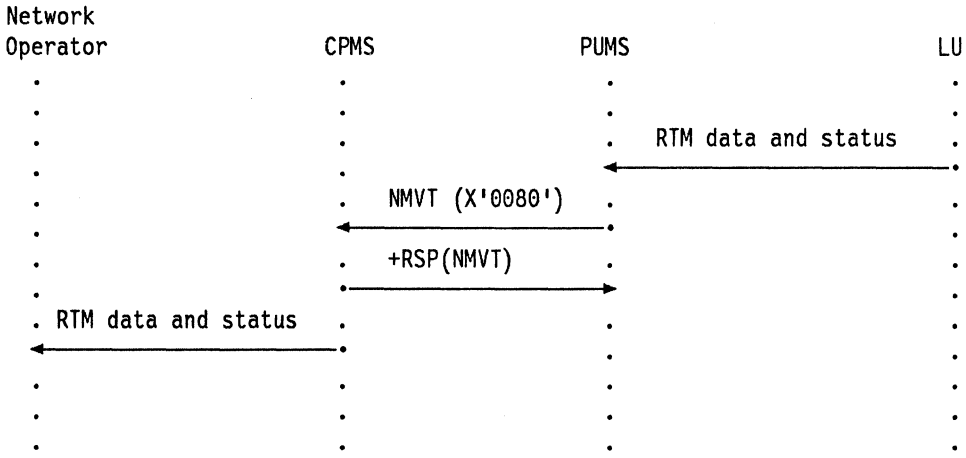


Figure 4-7. Example Flow Showing Unsolicited Return of RTM Data and Status

Chapter 5. Configuration Management

Configuration management is the control of information necessary to identify network resources. This physical identification includes information such as machine type and serial number (to identify hardware), program number and release level (to identify software), and port number and power-on status of port-attached devices. Query product ID is the only element of configuration management for which SNA management services have been provided.

Query Product ID (QPI)

Query Product ID (QPI) provides the capability to retrieve product-identification information from a specified PU. The information can be used to facilitate problem determination or problem diagnosis. Another use of the information is querying the identification of hardware and software products in a network for inventory purposes. This latter function is sometimes referred to as *Network Asset Management*.

Refer to Figure 5-1 for an overview of the steps required for QPI and the component that they are implemented within.

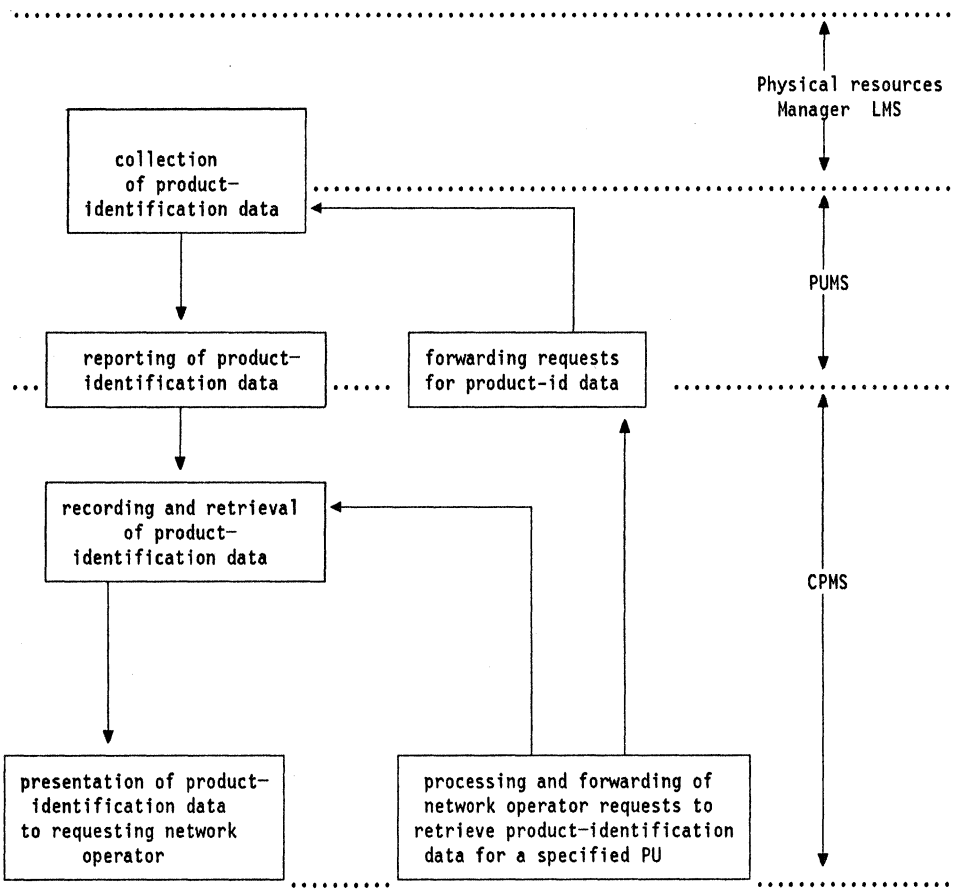


Figure 5-1. Overview and Placement of Query Product Identification (QPI) Steps

Functions Provided

QPI is composed of the following steps.

Collection of Product-Identification Data

The collection step of QPI provides both hardware and software (if present) identification for the specified PU and port-attached devices when requested.

- The following data is provided for each hardware product:
  - Machine type
  - Model number[1]
  - Serial number or repair ID number, consisting of a plant of manufacture identifier and a sequence number
  - Optionally, the engineering change (EC) level of the microcode in the product
  - Optionally, hardware product common name
  - Emulated product identifier, if the product emulates another hardware product
- The following data is provided for each software product:
  - Software product serviceable component identifier. This data is required for IBM software products assigned a component ID by the IBM National Service Division (NSD).
  - Software product program number. This field is required for IBM products not assigned a component ID by the IBM NSD.
  - Software product common level. This field is required for IBM products not assigned a component ID by the IBM NSD.
  - Software product common name
  - Either the software product customization identifier or the software product customization date and time, if the product is customer modifiable

The collection step of QPI provides the following configuration description for each port-attached device:

- The port number the device is attached to
- Whether the device is powered on or off
- Whether the device was powered on since the last QPI solicitation

---

[1] The model number must be provided if it is required, in conjunction with machine type and serial number to uniquely identify a product instance. If the model number is not needed to uniquely identify a product instance, its inclusion, for the purpose of providing additional information, is optional.

Forwarding Requests for Product-Identification data

This function requests product-identification data from the Physical Resources Manager LMS for the SNA node and port-attached devices.

Reporting of Product-Identification Data

The reporting step of QPI provides for the reporting of product-identification information from the specified PU to the requesting control point.

Recording and Retrieval of Product-Identification Data

The recording and retrieval step of QPI is implemented at the requesting control point and provides for data base management (e.g., storage, retrieval) of collected data.

Presentation of Product-Identification Data

The presentation step of QPI is implemented at the requesting control point and presents product-identification information to a requesting network operator.

Processing and Forwarding of Network Operator Requests

The processing and forwarding step of QPI is implemented at the requesting control point and provides a mechanism for accepting network operator requests for product-identification information, and forwarding those requests to the specified PU.

Implementation

Support of QPI is provided in a PU by "EP_QPI Function Set" on page 10-65. This is an optional function set implemented by type 2 nodes.

Format Usage

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Request Product Set ID (X'8090') major vector
Flow: From SSCP to PU (Normal)

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Reply Product Set ID (X'0090') major vector
Flow: From PU to SSCP (Normal)

A network operator can request product identification information for a specified PU. Data necessary for the request is passed to CPMS.

CPMS formats the request into an NMVT containing a Request Product Set ID (X'8090') major vector and sends it to PUMS. When PUMS receives this NMVT, it forwards the request to the Physical Resources Manager LMS. The LMS passes the requested data back to PUMS. PUMS formats the data into one or more NMVTs containing a Reply Product Set ID (X'0090') major vector and sends it to its controlling CPMS.

When CPMS receives the NMVT, it optionally logs the data and passes it to the network operator. The flow is of the type described in "Request/Reply Flows for Non-Bulk Data" on page 2-10. Refer to Figure 5-2 for an example flow.

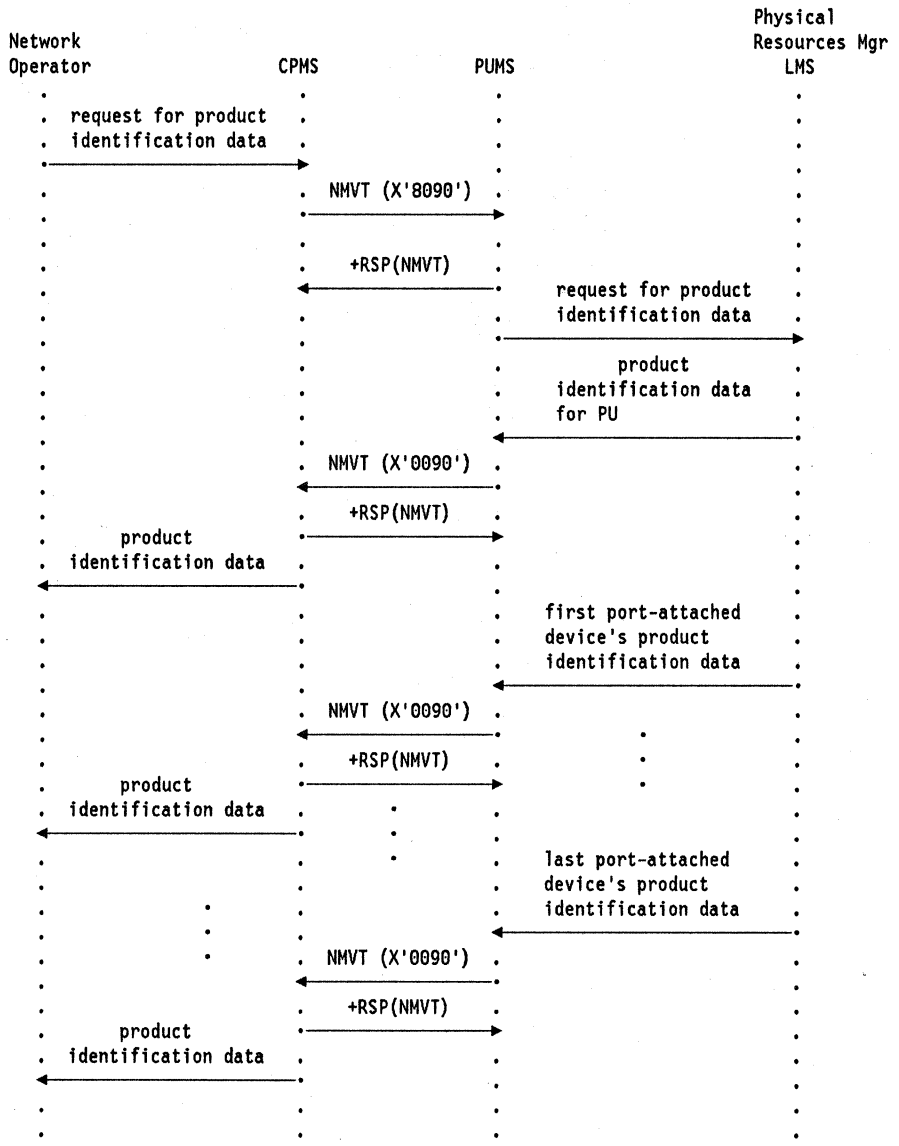
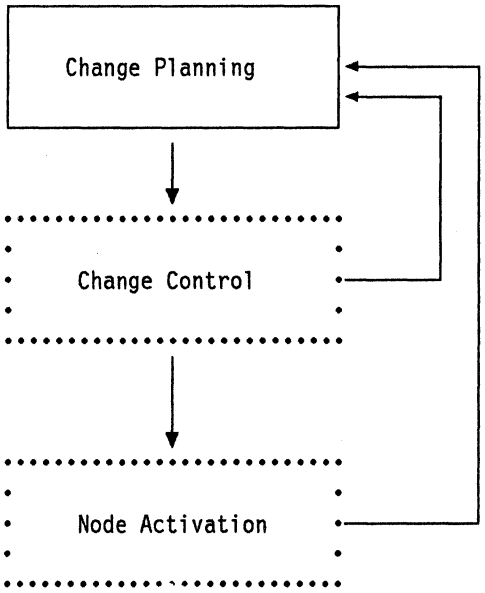


Figure 5-2. Example Flow Showing QPI Request to a PU and the Subsequent Replies. The flows beginning with the one labeled "first device's product identification data" occur only if the network operator queried for the identification of port-attached products as well as the products making up the SNA node.

Chapter 6. Change Management

Change management is the process of planning and controlling changes in a network. A *change* is defined as an alteration (addition, deletion, or modification) of one or more information system components, of one of the following types: hardware (may include microcode), or software (either system or application, vendor-supplied or user-written). Current architecture provides for management of changes that can be distributed electronically (not changes to hardware circuitry). For an overview of the elements of change management and their relationships to one another, refer to Figure 6-1.



Note: SNA management services are provided to assist with elements bounded by •••••.

Figure 6-1. Overview of Flow Between Change Management Elements

Change management, as described here and in subsequent chapters, defines the protocols followed and the formats that flow between nodes implementing this architecture. These descriptions are not intended to address a common user interface or programming interface.

Change Planning

Change planning is the element of change management that encompasses all the activities required to take place before changes can be distributed and installed. Refer to Figure 6-2 on page 6-4 for an overview of the steps required for change planning.

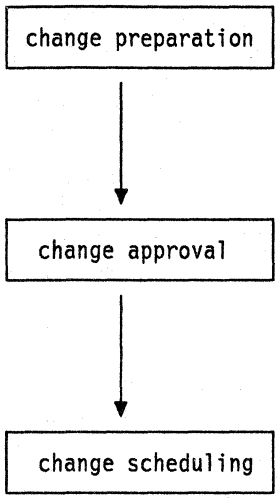


Figure 6-2. Overview of Change Planning Steps

Overview

A *network planner* is a person or program responsible for planning the configuration (and changes) of all or part of a network.

The need for changes to the network may arise for a variety of reasons. A non-exhaustive list of examples would include:

- Problem diagnosis identifies a required fix
- A vendor distributes preventive maintenance
- Users require configuration changes

When a network planner is aware of the need for a change to be made to the network, change planning begins. Change planning is composed of the following steps.

Change Preparation

For a general introduction to the SNA/MS *entry point* and *focal point* roles, refer to "Management Services Roles" on page 1-10.

Entry points are typically remote and unattended. An entry point may be a large system or a small one, an intelligent workstation, a fixed function device, or a control unit. One of the entry points may serve as the preparation site for *change files*, that is, files containing component replacements or updates and any necessary instructions to install them. Change files can contain one of the following types of data: microcode or microcode customizing data. The change file preparation entry point, if required, is typically located at the central site with the focal point, where it can be attended. The preparation site can also be at a focal point rather than at an entry point, or even in a system that serves neither a focal point nor an entry point role (in that case, the prepared change files must be introduced by some other means, for example, distribution tapes, at either a focal point or an entry point). It is the responsibility of the preparer

of the change file to include in it any necessary prerequisite information, to be checked by the target entry points when the change is installed. However, corequisites (a group of change files to be installed together) may be specified by the network planner.

Change Approval

Once a change file has been prepared for distribution and installation, approvals must be obtained from management. This may include various parts of the organization, such as network users and network management staff.

Change Scheduling

When change files have been approved for distribution and installation, the network planner can create a *change management plan*. The functions provided to the network planner that can be scheduled are described in more detail in "Change Control."

Change Control

Change control is the element of change management that distributes change files to entry points, and installs them there. Refer to Figure 6-3 for an overview of the steps required for change control.

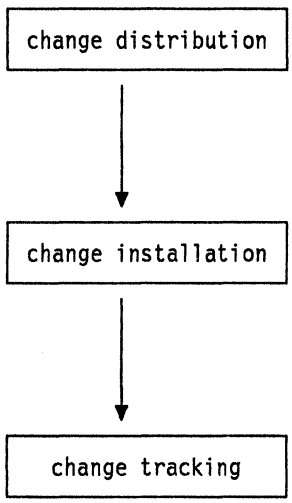


Figure 6-3. Overview of Change Control Steps

Overview

Change control is composed of the following steps.

Change Distribution

SNA/File Services (SNA/FS) architecture is used by management services for bulk data transport. SNA/FS moves files (that can be bulk data) from one location to another, provides a cross-enterprise-unique naming scheme (the SNA/FS *global name*), and defines the fetching and storing services. It uses enhanced SNA/Distribution Services formats, *format set 2*. For a detailed description of this architecture, refer to *SNA/File Services Reference, SC31-6807*, and *SNA/Distribution Services Reference, SC30-3098*.

SNA/MS defines a naming convention for change files. The name used is the SNA/FS global name, and SNA/MS specifies what values the tokens in that name can take.

SNA/MS specifies token values for microcode and microcode customization data only (not software, software customizing data, applications data, procedures, or documentation). However, the formats and protocols defined in MS are not sensitive to token values, and so are not restricted to microcode change files.

SNA/File Services capability is symmetric with respect to focal point and entry point roles. That is, nodes in either role can both send files to, and retrieve files from, other nodes. Change management commands, however, are issued only from a focal point. A summary of focal point and entry point roles with respect to change distribution and change management functions is provided in Table 6-1 on page 6-8.

SNA/DS is the transport mechanism used for the delivery of all commands and replies flowing between the focal point and entry point. Nodes between the focal point and the entry point may perform the SNA/DS intermediate role to provide a connectionless delivery service and fan-out (that is, one copy coming into the intermediate node can be replicated and forwarded to several subsequent nodes for ultimate routing to the destinations).

Change Installation

Once change files have been distributed to an entry point, they can be installed there. In fact, distribution and installation can be done with a single request if desired. See "Functions Provided" for a description of the installation functions.

Change Tracking

SNA/MS, SNA/FS, and SNA/DS reports are all tracked at the focal point. This allows a network planner to view the distribution and installation status and history for each entry point under control of the focal point.

Functions Provided

An SNA/MS change management focal point provides for the following change control requests at its interface with a network planner:

Retrieve: Obtains a change file prepared at an entry point or at another focal point, for storage at the focal point. If issued from an entry point, it obtains a change file from a focal point for storage at the entry point.

Send: Distributes a change file from the focal point to one or more entry points, or other focal points. If issued from an entry point, it distributes a change file from the entry point to one or more focal points, or other entry points.

Delete: Deletes a change file at one or more entry points, or other focal points.

Install: Uses a change file and its corequisites, if any, to alter, at one or more entry points, all components necessary to effect the change. The entry point can perform such alteration in a removable manner if requested, that is, so that a subsequent request (*Remove*) can return all those components to their condition prior to the alteration. The network planner can request testing, before and/or after the installation process is performed by the entry point. For example, an entry point can test a new version of a configuration file for validity before deleting the old version. Also, automatic removal of changes (if the tests or installation fail) or automatic acceptance (see *Accept* below) is possible.

The network planner can designate components altered by the installation process for *trial* activation, or alternately, *production* activation. This conditions how the entry point is later reactivated.

Send-and-Install: The same as Install, except that the focal point sends a change file in the same request.

Remove: Returns all components previously altered in connection with a change to their condition prior to the installation of the change. This is possible only for changes previously installed in a removable manner.

Accept: Relinquishes resources at an entry point required to maintain removability of a change. This cancels the removability of a change previously installed in a removable manner.

Focal point implementations may provide the network planner the ability to aggregate a series of requests, and specific scheduling and conditioning rules for their execution, into a *change management plan*.

Role Summary

The following is a summary of focal point and entry point roles with respect to the change distribution and change management functions described above.

Table 6-1. Functions and the Roles to Which They Apply			
Function	**FP Source EP Target**	**EP Source FP Target**	**FP Source FP Target**
RETRIEVE	YES	YES	YES
SEND	YES	YES	YES
DELETE	YES	NO	YES
INSTALL	YES	NO	NO
SEND_AND_INSTALL	YES	NO	NO
REMOVE	YES	NO	NO
ACCEPT	YES	NO	NO

Testing Features

Without the explicit testing features provided on the Install request, the only testing would be done by entry point user(s) over some period of time after the change was installed. If problems were encountered, the central-site network planner would need to be consulted, because only he would know about the previous installation of changes and be in a position to issue the Remove commands.

Two kinds of explicitly requested tests are required *as part of the installation process*, so that the central-site network planner can be informed immediately, as part of the installation report, about the results of certain diagnostic tests:

1. *Pre-tests* — Tests made before the programming components are altered
2. *Post-tests* — Tests made after the programming components are altered but before the installation report is made

Some reasons that automatic testing of changes is desirable:

- Possible corruption of the change between its initial development and installation at the entry point
- Possible sensitivity of the change to differences between the maintenance level of the target system and the maintenance level of those systems where the change has already been tested
- Possible sensitivity of the change to differences between the configuration of the target system and the configuration of those systems where the change has already been tested
- Possible sensitivity of the change to differences between functions or applications present at the target system and those present at systems where the change has already been tested

The *pre-test* is performed (if requested) by the entry point by examining the change file *before* components are altered. The change file contents may be examined for self-consistency, and consistency with the entry point's configuration, maintenance level, or application set. For example, an altered version of a file that contains input data to a routine can be checked to see if it contains inconsistent specifications.

If the pre-test fails, then no attempt is made to alter the components and the installation report is made to the requester with the test results.

The *post-test* is performed (if requested) by the entry point *after* components are altered, but before the installation report is made. Altered versions of the components are tested directly — for example, by executing diagnostic routines. Such routines or test instructions can be distributed along with the change.

If the post-test fails, then the components are returned to their unaltered condition if another parameter, *automatic removal*, is specified. In any event, the installation report is sent with the test results, and the requester is informed immediately. In this way, the requester has the opportunity to remove the change so that the impact on end users is minimized.

Implementations may choose to include testing procedures in the change files, so that the general procedures do not have to be developed and included in advance at the target entry points.

Implementation

Support of controlling changes is provided in an entry point by "EP_CHANGE_MGMT Function Set" on page 10-71. This is an optional function set.

A series of example change mangement flows is provided in "Example Flows" on page 10-82. These examples show, in detail, the data that flows between the focal point and entry point as well as the interaction between SNA/MS, SNA/DS and SNA/FS at each node.

Format Usage

SNA/FS AGENT OBJECT
  FS_Agent_Request GDS Variable (X'1530')
  FS_Agent_Report GDS Variable (X'154A')
  Flow: From CP to PU (over LU-LU sessions used by SNA/DS)

This format is used for the Retrieve, Send and Delete functions which exploit the FS-capability of the SNA/MS agent. A CP-MSU does not flow in the agent object.

Figure 6-4 on page 6-10 shows an example flow for the retrieve function. The flow is of the type described in "Request/Reply Flows for Bulk Data" on page 2-13. The flow consists of the following stages:

1. A network planner requests the focal point SNA/MS to retrieve a change file from an entry point. The request is passed to SNA/MS in the focal point.
2. The focal point SNA/MS formats an SNA/FS request for the entry point to fetch and return a change file. The focal point sends the command to the entry point.

3. SNA/MS passes the request to the SNA/FS server.
4. The SNA/FS server fetches the requested change file and sends it to the focal point SNA/MS on a SNA/DS conversation.
5. The SNA/FS server at the focal point stores the change file and SNA/DS notifies SNA/MS that it has arrived.
6. SNA/MS notifies the network planner that the change file has arrived.

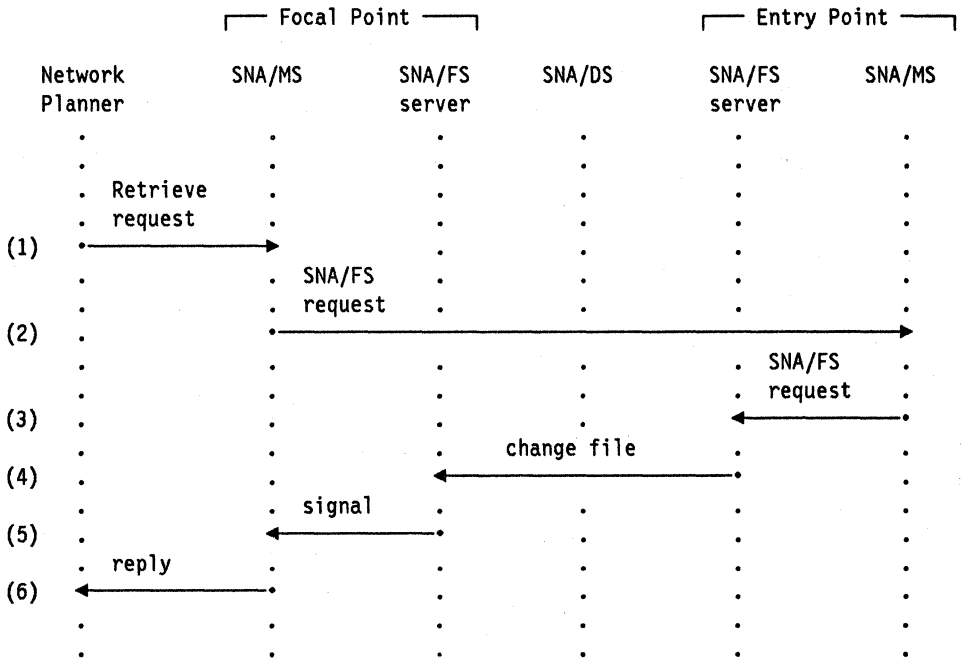


Figure 6-4. The Request/Reply Flow for Retrieving Change Files

CONTROL POINT MANAGEMENT SERVICES UNIT (CP-MSU)
Request Change Control MV (X'8050')
Change Control MV (X'0050')
Flow: From CP to PU (over LU-LU sessions used by SNA/DS)

This format is used for the Install, Remove and Accept functions. A CP-MSU containing the appropriate SNA/MS major vector flows in the agent object.

Figure 6-5 on page 6-11 shows a request/reply flow to send a change to an entry point and install it there. The flow is of the type described in "Request/Reply Flows for Bulk Data" on page 2-13. The flow consists of the following stages:

1. A network planner requests that a change be sent to an entry point. An Install function is also requested. The planner's request is passed to SNA/MS in the focal point.
2. The focal point SNA/MS formats a request CP-MSU containing a Request Change Control (X'8050') major vector. It invokes its SNA/FS server to fetch the change file from storage.
3. The SNA/FS server fetches the file, and it flows together with the CP-MSU on a SNA/DS conversation between the focal point and the entry point.
4. The SNA/FS server in the entry point stores the change file and SNA/DS notifies SNA/MS that it has arrived, passing the CP-MSU and SNA/FS parameters in the agent object.
5. SNA/MS installs the change file, and builds a reply CP-MSU containing a Change Control (X'0050') major vector. Then it sends the report to the focal point SNA/MS on a SNA/DS conversation.
6. After the report has reached the focal point SNA/MS, it is passed to the network planner that made the request.

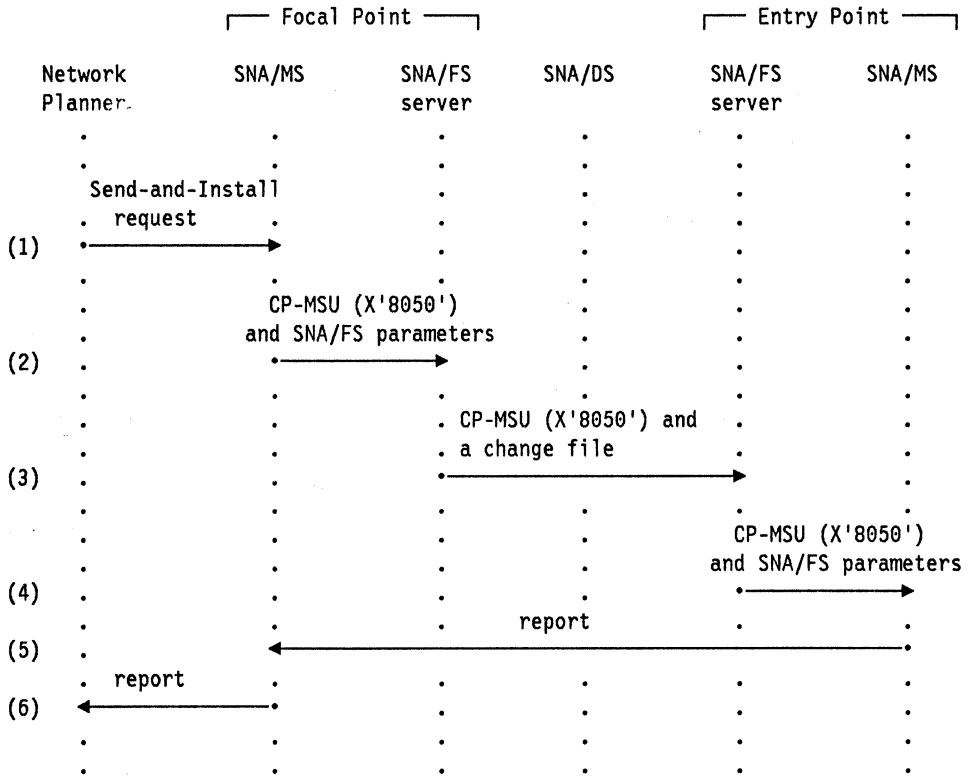


Figure 6-5. The Request/Reply Flow for Send-and-Install

Figure 6-6 on page 6-12 shows a request/reply flow to remove a change already installed removably at an entry point. The flow is of the type described in "Request/Reply Flows for Bulk Data" on page 2-13. The flow consists of the following stages:

1. A network planner requests that a change be removed at an entry point. The planner's request is passed to SNA/MS in the focal point.
2. The focal point SNA/MS formats a request CP-MSU containing a Request Change Control (X'8050') major vector. The SNA/FS server is invoked to build the SNA/FS control information to identify the change file to be removed.
3. The CP-MSU and control information is sent on an SNA/DS conversation between the focal point and the entry point.
4. The SNA/FS server in the entry point decodes the control information, and SNA/DS notifies SNA/MS that the request has arrived, passing the CP-MSU and SNA/FS parameters in the agent object.
5. SNA/MS removes the change file, and builds a reply CP-MSU containing a Change Control (X'0050') major vector. Then it sends the report to the focal point SNA/MS on a SNA/DS conversation.
6. After the report has reached the focal point SNA/MS, it is passed to the network planner that made the request.

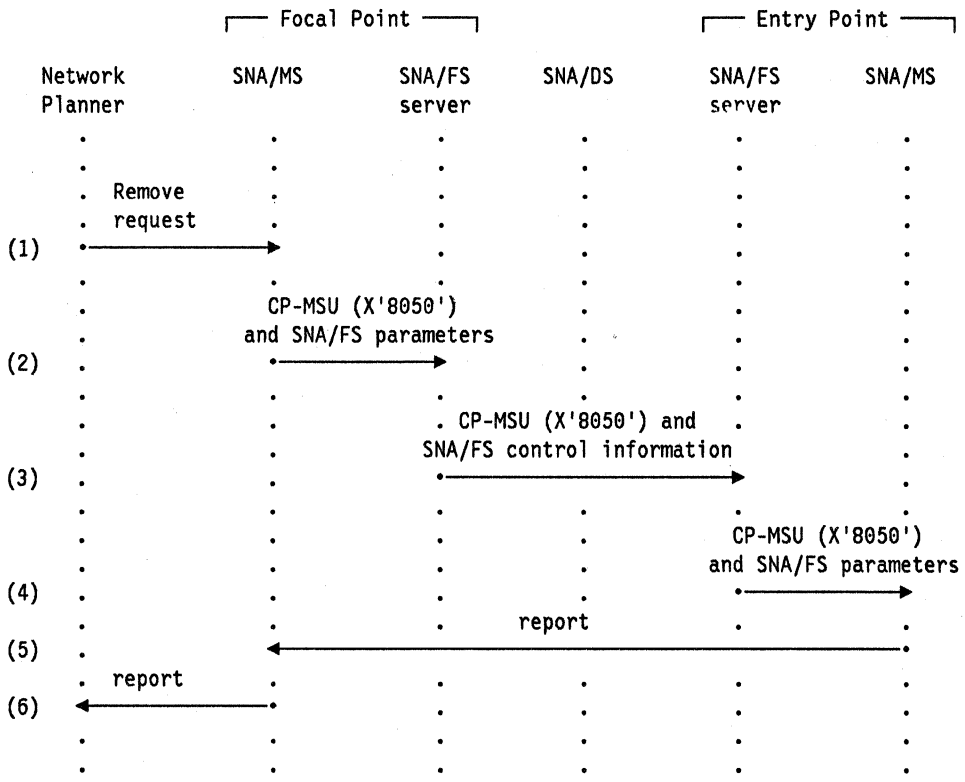


Figure 6-6. The Request/Reply Flow for Remove

Other key examples (similar to that just described for Remove, and so not separately shown) include:

- An Install request acting on a change file that was sent previously
- An Accept request

Node Activation

Functions Provided

An MS change management focal point provides for the following request at its protocol boundary with the network planner:

Activate: Causes reactivation of an entry point. Such reactivation uses changes installed on a trial basis in addition to those installed in production. In addition, the network planner can request that the entry point not attempt reactivation if user sessions are currently active at or through the entry point.

The Activate command provides a network planner the ability to reactivate an entry point in a change management plan. For example, an initial microprogram load (IML) of the node may be performed. A reply is returned by the entry point prior to reactivation, to indicate acceptance of the Activate command.

Unlike other change management functions, Activate does not refer to bulk data and so the SNA/File Services server is not invoked by SNA/Distribution Services in the process of distributing requests and reports.

The Activation Use Parameter

The *activation use* parameter of the Install request causes the entry point to install the indicated changes for *trial activation* or *production activation*. The Activate request contains a parameter that causes the entry point to activate both the trial and the production versions of altered entry point components. Entry points implement both of the following types of local reactivation:

1. Use of both trial and production components, and
2. Use of production components only.

Changes that cannot be tested fully or that have a strong potential to affect the entry point to focal point communication path are best installed on trial. After activation and a period of successful operation, the changes may be installed again in production.

Of course, to provide the trial activation capability, an entry point must be capable of installing a change removably. That is, the entry point must be able to keep a copy of the production-level system. Storage capabilities at the entry points may preclude support of trial activation in some cases. If so, the installation will be refused if activation use of *trial* is specified.

The parameters on the *Install* and *Activate* requests reflect very specific reliability requirements for entry point implementations. While the basics of the Activate request provide the ability to reactivate an entry point after changes have been installed (for example, by loading altered microcode), there is a need to provide a way to use only unaltered versions of components during local activation. Without such capability, reactivation of the entry point could result in the use of a change so destructive that the path to the focal point cannot be maintained. Repair (in the form of further change distribution and installation) cannot be triggered by the focal point in this case. Reactivation

must be triggered at the entry point through human intervention. Reactivation of the (working) production level of components can restore communication with the focal point and allow the network planner to repair the components.

Hence, the ability to store both a production and a trial set of entry point components is necessary to allow operator intervention to be simple, and to avoid the requirement for both change management skills and awareness of network planning activities at the entry point. Through support of a default local activation of the production system, an entry point implementation can provide hardware externals that are very simple for non-technical users of entry points.

Implementation

Support of activating nodes is provided in an entry point by "EP_CHANGE_MGMT Function Set" on page 10-71. This is an optional function set.

Format Usage

CONTROL POINT MANAGEMENT SERVICES UNIT (CP-MSU)
Request Activation MV (X'8066')
Activation Acceptance MV (X'0066')
Flow: From CP to PU (over LU-LU sessions used by SNA/DS)

This format is used for the Activate function. A CP-MSU containing the appropriate SNA/MS major vector flows in the agent object.

Figure 6-7 on page 6-15 shows an example flow for the activate function. The flow is of the type described in "Request/Reply Flows for Bulk Data" on page 2-13. The flow consists of the following stages:

1. A network planner requests the focal point SNA/MS to activate an entry point.
2. The focal point SNA/MS formats a request CP-MSU. The focal point then invokes SNA/DS to send the CP-MSU on a conversation between the focal point and the entry point.
3. SNA/MS builds a CP-MSU indicating that activation of the entry point will be performed and sends the CP-MSU to the focal point.
4. SNA/MS indicates activation acceptance to the network planner. SNA/MS causes the entry point (including SNA/MS itself) to be reactivated, and the LU-LU session with the focal point is terminated.

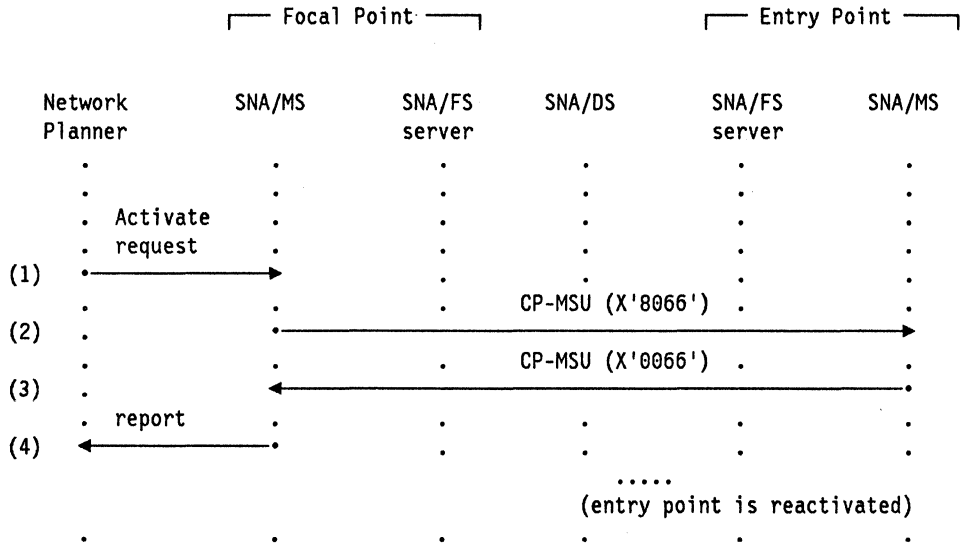


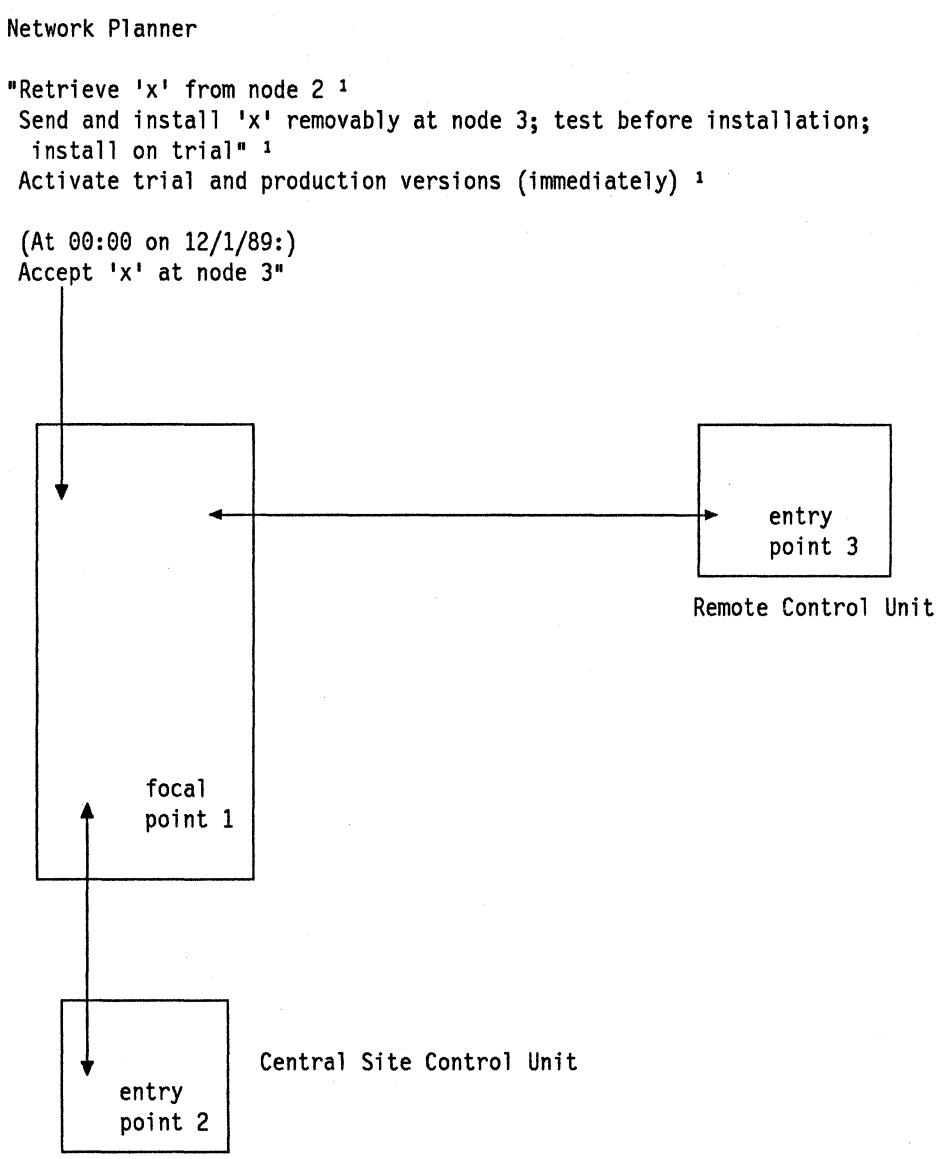
Figure 6-7. The Request/Reply Flow for Activate

Example Change Management Plan

Centralized Customization of Microcode

Refer to Figure 6-8.

A network planner at focal point 1 wants to retrieve a microcode customizing data file that was developed at a central site control unit (entry point 2), then send, test, and install it at a remote control unit (entry point 3), and (one week later) accept it.



[1] If this is not successful, the next function will not be performed and the planner is notified that the plan ended

Figure 6-8. Centralized Customization of Microcode

Chapter 7. Common Operations Services

Common operations services are a set of services available to all of the major categories of network management. The architecture for common operations services provides a way of managing types of resources not explicitly addressed by the architecture defined for the individual categories. It does this by providing a general mechanism that allows a network operator to communicate with specialized network management applications; these applications, in turn, provide functions not currently provided by SNA management services.

Common Operations Services for Resource Control

A specific set of common operations services has been defined to assist a network operator with the task of controlling resources in the network. In order to manage a network, an operator requires the ability both to know about the various resources present in the network and to control those resources. Much of the necessary information about network resources is provided by the unsolicited messages defined by the management services architecture, e.g., the Alert. Other information is available via architecturally-defined requests, such as QPI.

The common operations services for resource control provide for the collection of the additional information necessary to operate a network. They also provide a mechanism by which an operator can exert control over network resources, by directing commands to them.

The architecture for common operations services defines a number of capabilities to be used by applications that provide network management services. The structures defined by this architecture serve network management applications distributed in a network. Their role is to facilitate the transfer of information between two portions of a network management application, or between a network operator and a remote network management application. In this way it is possible for functions and/or resources not explicitly covered by the management services architecture to be brought within its overall network management scheme.

Functions Provided

The common operations services structures for resource control provide for the transport of information between network management applications, or between a network operator and one such application. The meaning of the information being transported is entirely defined by the applications using the supporting services. The common operations services architecture provides, at the highest level, for two functions:

- The enveloping of uninterpreted data for transport through an SNA network
- A means of identifying a particular network management application, or a particular (human or programmed) network operator, at the destination node as the intended recipient of this data.

Enveloping of Data

Two of the structures defined by common operations services for resource control are entirely free-form. The data provided by the served applications is enveloped within subvectors that identify how the data is encoded (e.g., as character data in a specified coded graphic character set), but say nothing about the nature of the data. The data is then transported through the management services transport network, on the same sessions that carry such messages as Alerts.

The other three structures defined by common operations services identify the nature of the data they transport (for example, a query for information about a resource), as well as how it is encoded.

Routing to Served Applications and Network Operators

When a network operator enters a common operations services command, the name of a destination application is also specified. Similarly, when an application sends a message to a network operator, it also identifies the operator. Since the network operator in these cases can be programmed as well as human, the capability is provided for a network management application at a host to communicate with a network management application in the network, and vice versa.

See "Routing a Request to a Specified Application" on page 8-15 for a complete discussion of this capability.

Parameter Major Vectors

As noted earlier in "Introduction to Management Services Formats," the common operations services architecture for resource control exploits the capability of the NMVT to transport multiple major vectors. In addition to the usual management services major vector, an NMVT defined for common operations services for resource control contains one or more *management services parameter* major vectors, which appear after the usual major vector in the NMVT. MS parameter major vectors contain and group subvectors that carry the data associated with a single resource, or they indicate resource groupings. Figure 7-1 on page 7-5 illustrates the two ways in which parameter major vectors are used by common operations services.

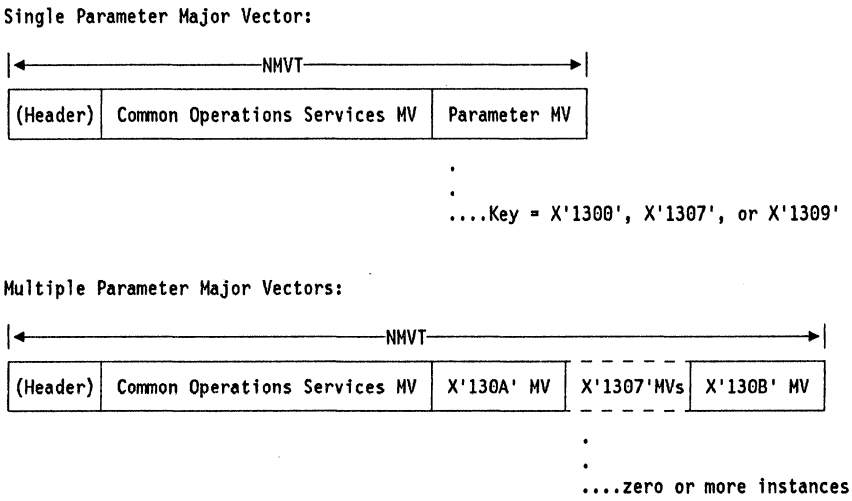


Figure 7-1. Parameter Major Vectors in Common Operations Services Encodings

Three MS parameter major vectors have been defined to handle the three types of data to be carried:

- Character strings (the Text Data major vector)
- Identified values (the Structured Data major vector)
- An undefined format (the Transparent Coded Datastream major vector).

Unlike other major vectors, the last of these MS parameter major vectors has no subvectors defined for it. The definition of its content is left totally up to the applications using it.

The *Resource Data* subvector serves as the general carrier of identified values within the Structured Data major vector. This subvector carries a name identifier and one of several defined types of data (character, integer, hexadecimal or bit-string) for each separately identified item of data reported for the resource.

Two additional MS parameter major vectors have been defined, to allow the limits of the set of MS parameter major vectors within a single NMVT to be identified. The Begin Data Parameters MS parameter major vector is used to indicate that one or more Structured Data MS parameter major vectors may follow. Depending on what common operations services major vector it accompanies, the Begin Data Parameters may or may not have any Structured Data MS parameter major vectors following it. The End Parameter Data MS parameter major vector terminates all MS parameter major vectors. The use of these two parameter major vectors is illustrated in Figure 7-1.

Commands

The Execute Command MS major vector provides for transport of a command from a human or programmed network operator to a specified network management application. The Reply to Execute Command major vector carries one of three MS parameter major vectors, and is routed to the originator of the Execute Command.

Messages to an Operator

The Send Message to Operator MS major vector provides for transport of a message from a network management application to a specified (human or programmed) network operator. Its function is thus complementary to that of the Execute Command major vector. One of three MS parameter major vectors is carried with the Send Message to Operator major vector.

Querying for Information on a Resource

The explicit request for information relating to a named resource (or multiple named resources) is provided in the Query Resource Data MS major vector. The request always contains the name of one or more resources.

The Reply to Query Resource Data MS major vector is followed by the Begin Data Parameters and End Parameter Data pair of MS parameter major vectors and one Structured Data MS parameter major vector for each resource.

Testing a Resource

The Test Resource MS major vector provides the same list of names as Query Resource Data, plus a Test Request subvector. Reply to Test Resource uses the same MS parameter major vector structure for reporting information as does Reply to Query Resource Data. The Reply to Test Resource MS major vector carries additional information in the Test Result subvector.

The only type of test identified in the request is a "Self Test." The number of times the test is executed may be specified. A summary indication of the results of the test is returned, along with detailed information on each resource included in the test.

Analysis of a Resource

The Analyze Status MS major vector is very specifically an extension of the Alert function. It uses the same list of resource names in the request as the Query Resource Data and Test Resource major vectors, and returns a Structured Data MS parameter major vector series similar to those returned by the Reply to Query Resource Data and Reply to Test Resource major vectors, but without the detailed data. Instead, the Reply to Analyze Status major vector returns a Probable Causes subvector identical to that found in an Alert. The Probable Causes subvector is returned in the Begin Data Parameters MS parameter major vector, and applies to the entire set of resources identified in the request. The Structured Data major vectors for each resource then identify the individual resources by name and type.

The Analyze Status MS major vector is sent to an application that manages a resource suspected of failure, but for which there has been no Alert received.

Implementation

Support of common operations services for resource control is provided in a PU by "EP_COMMON_OPERATIONS_SERVICES Function Set" on page 10-138. This is an optional function set implemented by type 2 nodes.

Format Usage

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Execute Command MV (X'8061')
Analyze Status MV (X'8062')
Query Resource Data MV (X'8063')
Test Resource MV (X'8064')
Flow: From SSCP to PU T2 (Normal)

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT)
Reply to Execute Command MV (X'0061')
Reply to Analyze Status MV (X'0062')
Reply to Query Resource Data MV (X'0063')
Reply to Test Resource MV (X'0064')
Send Message to Operator MV (X'006F')
Text Data Parameter MV (X'1300')
Structured Data Parameter MV (X'1307')
Transparent Coded Datastream Parameter MV (X'1309')
Begin Data Parameters Parameter MV (X'130A')
End Parameter Data Parameter MV (X'130B')
Flow: From PU T2 to SSCP (Normal)

A network operator can send a common operations services command to a specified served application at a specified PU. Data necessary for creating the request is passed to CPMS.

CPMS formats the request into one of four common operations services major vectors:

- Execute Command (X'8061')
- Analyze Status (X'8062')
- Query Resource Data (X'8063')
- Test Resource (X'8064').

CPMS then constructs an NMVT containing the major vector, and sends it to PUMS.

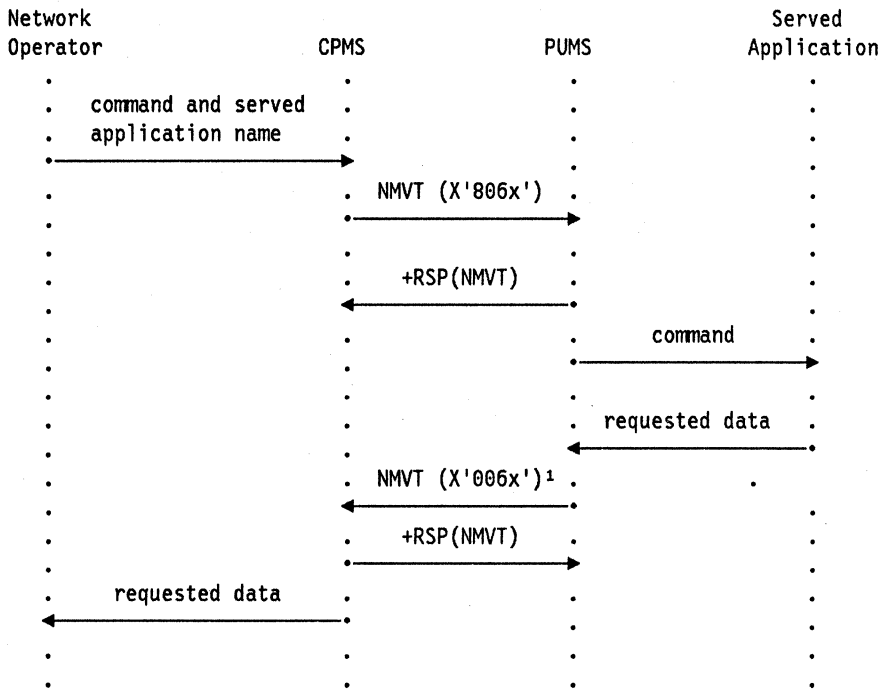
When PUMS receives the request, it passes it to the served application identified in the Name List (X'06') subvector.

After performing the requested function, the application to which the request was passed creates one of the following four common operations services major vectors, as appropriate, in response to the request:

- Reply to Execute Command (X'0061')
- Reply to Analyze Status (X'0062')
- Reply to Query Resource Data (X'0063')

- Reply to Test Resource (X'0064').

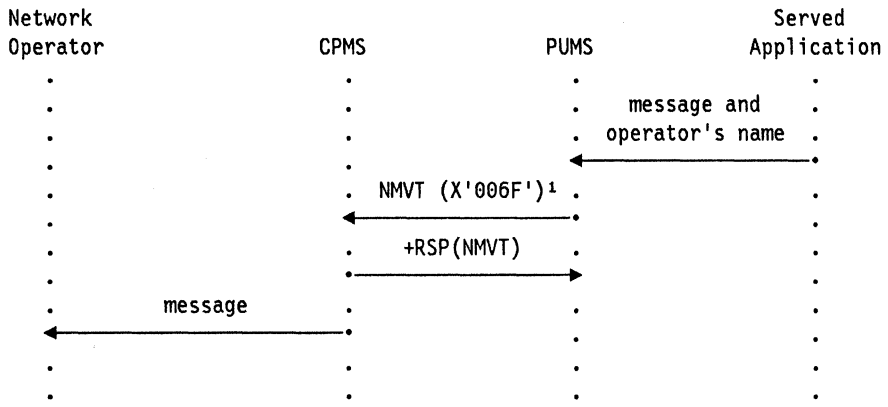
It also creates one or more parameter major vectors. The application passes the major vectors it has created to PUMS. PUMS envelopes the major vector in an NMVT and sends it to CPMS. CPMS optionally logs the reply and passes it to the network operator. The flow is of the type described in "Request/Reply Flows for Non-Bulk Data" on page 2-10. Refer to Figure 7-2 for an example flow.



[1]Plus one or more parameter major vectors.

Figure 7-2. Example Flow Showing Common Operations Services Request and Reply

A served application can also send a text message to a network operator at a specified node. It builds a Send Message to Operator (X'006F') major vector, plus one of three parameter major vectors (Text Data (X'1300'), Structured Data (X'1307'), or Transparent Coded Datastream (X'1309')); then it passes these major vectors to PUMS in its node. PUMS envelopes the major vector in an NMVT and sends it to CPMS. CPMS optionally logs the message and passes it to the network operator. The network operator then passes the message to the individually-identified operator to whom it was directed. The flow is of the type described in "Unsolicited Flows" on page 2-8. Refer to Figure 7-3 on page 7-9 for an example flow.



[1]Plus one parameter major vector.

Figure 7-3. Example Flow Showing Common Operations Services Send Message to Operator

Part 2. Architectural Logic for Management Services

Part II

Chapter 8. Overview of the Architectural Logic for Management Services

Transport of Management Services Data

The following sections describe the options available to management services applications for transporting management services data. The SSCP-PU session is used for transferring management services data between a control point and a physical unit. The characteristics of this session are described briefly in "Transport of Management Services Data on the SSCP-PU Session."

The Change Management category uses SNA/File Services and SNA/Distribution Services for distribution of potentially large files, requests to manipulate them, and reports to track the distribution and installation. The management services transport for bulk data is described in "Transport of Bulk Management Services Data" on page 8-4.

Transport of Management Services Data on the SSCP-PU Session

The primary path for transport of SNA/MS data between nodes is the SSCP-PU session. SNA/MS plays no role in the establishment of this session. Since the session is established when a PU is activated, it is already present when PUMS comes up. From the point of view of PUMS, the SSCP-PU session is simply a pipe through which management services requests and data can be exchanged with the PU's controlling SSCP.

The profiles for the SSCP-PU session are documented in *SNA Formats*. The SSCP-PU session operates according to TS profile 1 and FM profile 0. An important property of the session for SNA/MS is the maximum RU size: 256 bytes outbound (i.e., from the SSCP to the PU) and 512 bytes inbound (from the PU to the SSCP). See *SNA Formats* for further properties of the SSCP-PU session.

Transport of Bulk Management Services Data

This section describes how management services uses SNA/DS for transport of requests, reports, and bulk data.

Introduction to SNA/DS and SNA/FS

Refer to Figure 8-1.

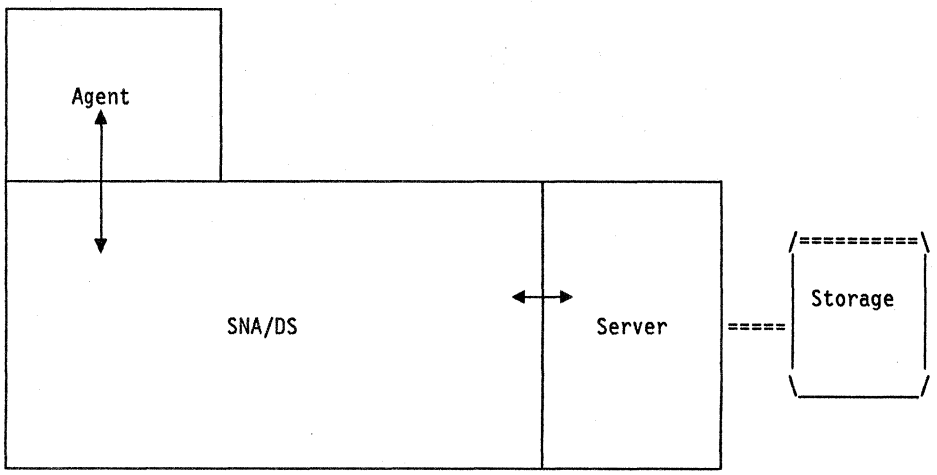


Figure 8-1. SNA/DS View of Agent and Server

In SNA/DS, the *agent* is an application transaction program that is using SNA/DS as a transport mechanism.

The *server*, also provided by the user of SNA/DS, is invoked by SNA/DS to handle staging and destaging of (typically large) files to the system storage facilities.

The SNA/FS Server

For network management, the server is involved in distributing large files, but not administering them. In fact, the server cannot distinguish between a network management file and, say, a job to be submitted for execution, or a file simply to be stored with no further processing. Such actions are the responsibility of the agent.

The building and parsing of the object handled by the server (the server object) for network management need be no different from that for other SNA/DS agents. For this reason, architecture has been developed for the server, called *SNA/File Services* (SNA/FS).

For a detailed description of SNA/File Services, refer to *SNA/File Services Reference, SC31-6807*.

How Management Services Uses SNA/FS

A user SNA/MS request is converted by a focal point or entry point into another kind of request, called a *command*, that it sends to one or more entry points or focal points. For example, the Retrieve request is converted to an SNA/FS *Transfer-to-Requester* command. A command is executed by the target node, and results in a *report* to the requester. Each report, associated with a request, is given to the requesting user when it is received by the requesting node. Reports are defined by each of the SNA components. An SNA/DS report is received if an error occurred in the distribution network. If distribution is successful, then either an SNA/FS report or an SNA/MS report is received. SNA/FS reports occur if the request was for file transfer only, or for SNA/MS requests that resulted in unsuccessful file transfer. If file transfer succeeds for an SNA/MS request, then SNA/MS reports indicate the success or failure of the SNA/MS request.

Commands and reports are carried in SNA/DS *message units* (MU)s.

Refer to "The Management Services Formats" on page 2-3 for a description of the MU format and a pictorial representation of the MU.

Each MU may contain the following items:

- A command or report, defined by either SNA/MS or SNA/FS, and contained in the agent object;
- SNA/FS control information contained in the server object; and
- A file, also contained in the server object.

An MU may identify, but not carry, a file, in which case a SNA/FS control information is present without a file. An example of this is a *Retrieve* request. On the other hand, a command or report need not be present. For example, a successful Retrieve request results in the return of a file and its control information, but no report.

Different from both user requests and focal point commands commands is the SNA/FS *server instruction* in the control information. The server instruction indicates to the SNA/FS server at the destination node how to manipulate the file into, or out of, the storage facilities. For example, a Transfer-to-Requester command flows with a *Fetch* server instruction, and the reply contains a *Create/Load-or-Replace* server instruction.

Example of Data Flow:

We shall follow a step-by-step sequence of events, starting with a typical example of a request, issued by a user at a focal point.

The SNA/FS capability is symmetric with respect to focal point and entry point, so this request could also originate at an entry point, in the case of a request that involved file transfer only (and not some additional action such as change management).

First, the request is issued by the user, and contains the agent unit-of-work correlator, SNA/FS control information (including the file name), and a list of destinations.

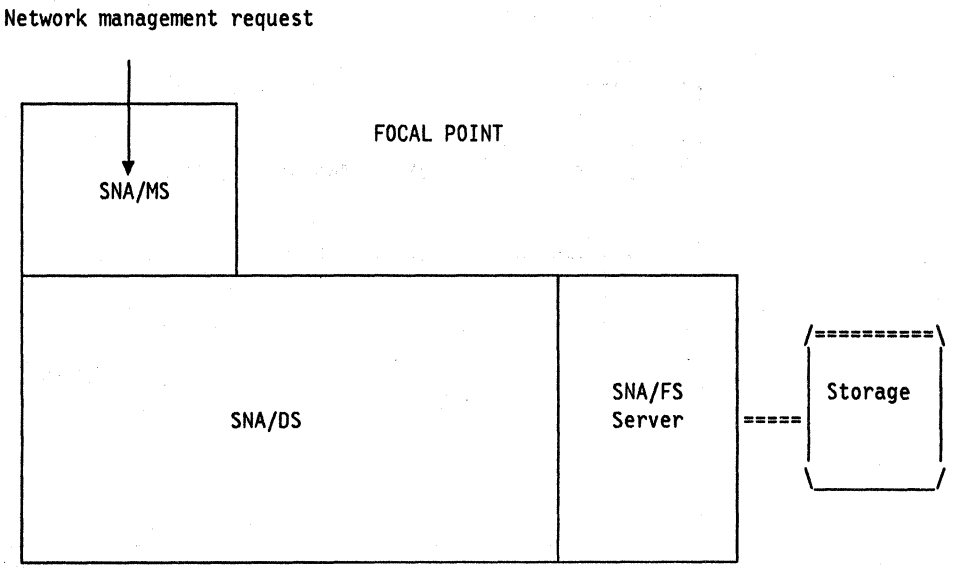


Figure 8-2. Example Flow: User Issues Request

Next, SNA/MS builds the agent object, containing the command to the entry point. The SNA/DS SEND_DISTRIBUTION verb is issued, specifying the agent object, the correlator, the destination list, the destination agent name (SNA/MS), the destination server name (SNA/FS), and the SNA/FS server parameters.

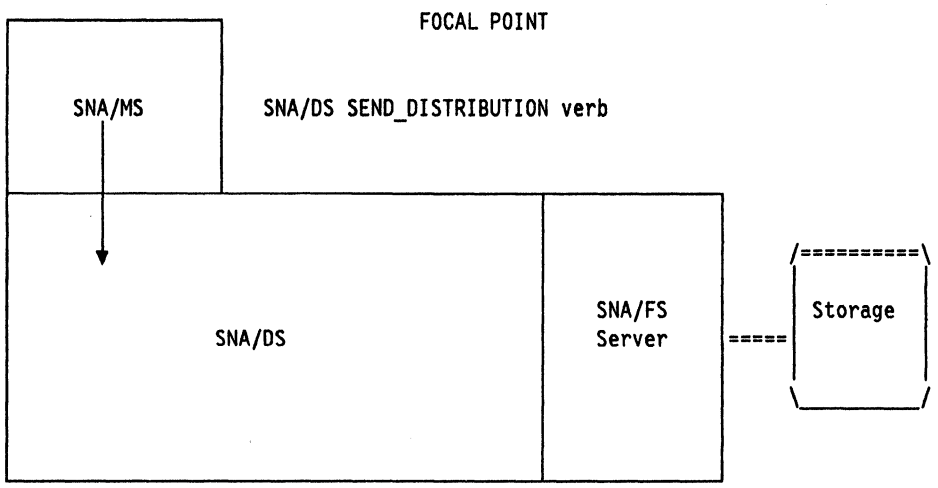


Figure 8-3. Example Flow: SNA/MS Builds Agent Object and Issues SEND_DISTRIBUTION

For each destination as appropriate, SNA/DS allocates a conversation, builds an SNA/DS MU) with the help of the SNA/FS server, and sends the MU on that conversation. The MU is pictured here.

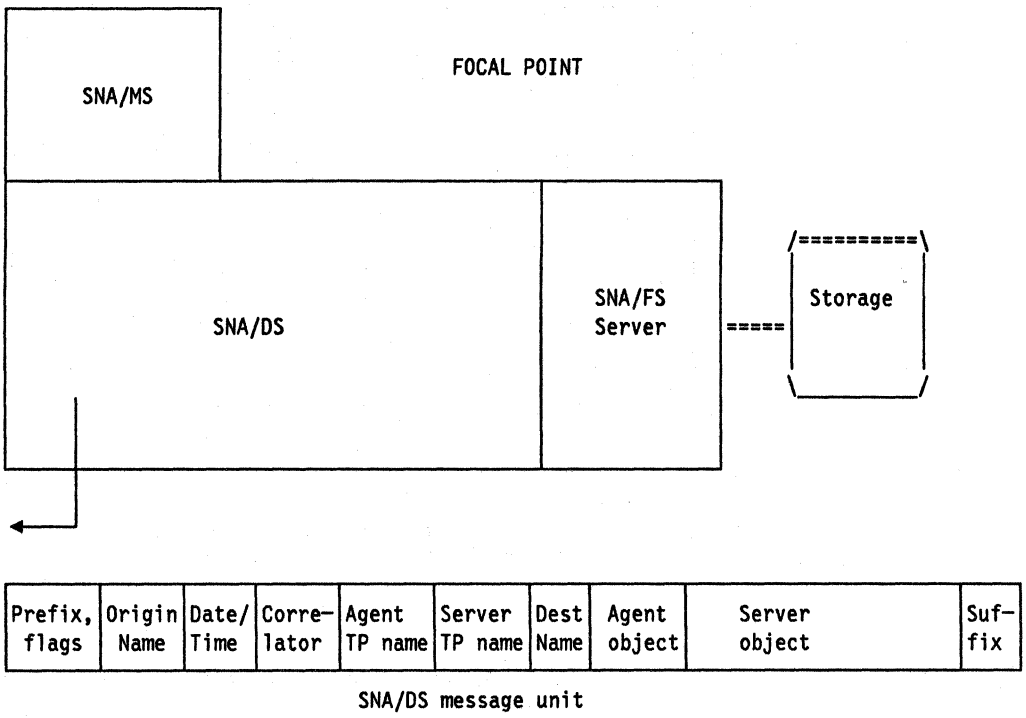


Figure 8-4. Example Flow: SNA/DS Builds a Message Unit and Sends It to the Entry Point

At the destination node, SNA/DS receives and decodes the MU with the help of the SNA/FS server. The server interprets the SNA/FS control information and stores the file.

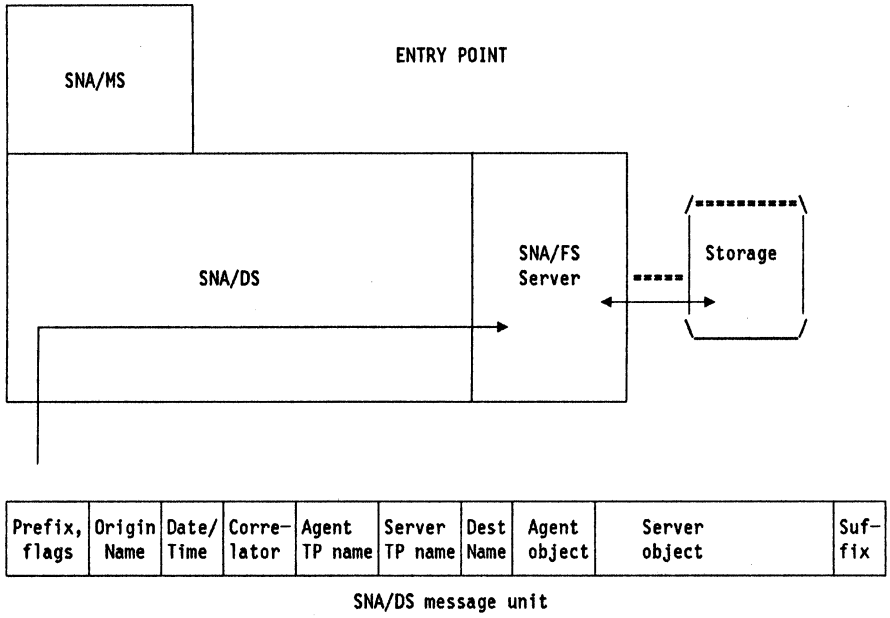


Figure 8-5. Example Flow: SNA/DS at the Entry Point Receives the MU and Invokes the SNA/FS Server

The SNA/MS agent at the entry point issues RECEIVE_DISTRIBUTION and is returned parameters, including: the correlator value, the agent object, and the SNA/FS parameters.

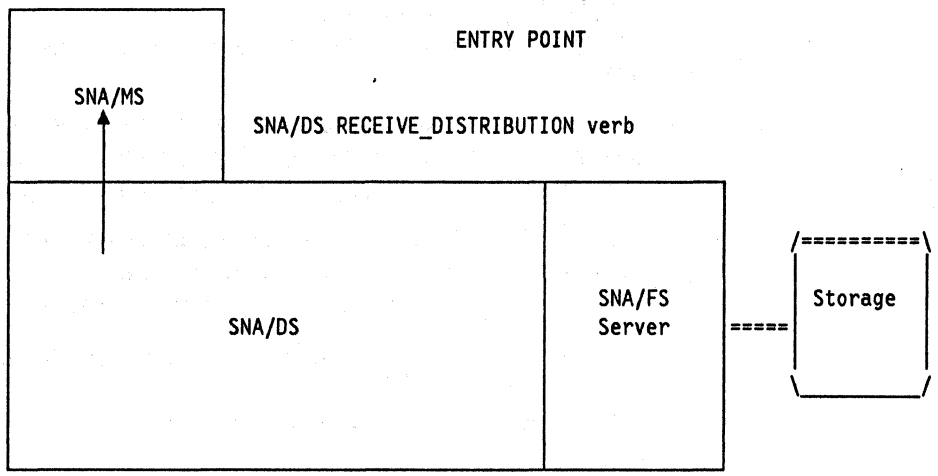


Figure 8-6. Example Flow: SNA/MS at the Entry Point is Returned RECEIVE_DISTRIBUTION Parameters

SNA/MS then parses the agent object, interprets the results of the SNA/FS server instruction, and executes the intended command.

The entry point returns (echoes) the correlator value supplied by the focal point.

The return flow is similar, except that at the focal point, a reply is given to the issuer of the request.

More detailed examples are provided in "EP_CHANGE_MGMT Function Set" on page 10-71.

SNA/FS Commands and Server Instructions Used by SNA/MS:

Management services uses the following SNA/FS commands:

- TRANSFER_TO_REQUESTER - The agent in the target node is requested to return one or more files.
- REPORT_FS_ACTION - The other agent is requested to report whether or not its server executed the server instruction successfully.
- REPORTING_FS_ACTION - The other agent is being informed whether or not a server instruction was executed successfully.

The agent object is omitted in the reply to a TRANSFER_TO_REQUESTER.

Management services uses the following SNA/FS server instructions:

- CREATE/LOAD_OR_REPLACE - This requires the SNA/FS server to determine whether or not the file received already exists on its system. If it does, the server replaces it. If it doesn't the server creates a new file.
- CREATE/LOAD - This requires the SNA/FS server to accept the contents of the server object and store it in a new file.
- DELETE - This requires the SNA/FS server to locate the identified object and delete it.
- ENCODE_ONLY - This requires that the SNA/FS server encode (build) the server object and then pass the result to SNA/DS. The server object does not contain a file, just file control information, so no fetching is required.
- DECODE_ONLY - This requires that the SNA/FS server decode (parse) the identifier and then pass the result to the agent. The server object does not contain a file, just file control information, so no storing is required.
- FETCH - This requires the source server to fetch the files from storage.

Refer to Table 8-1 for a summary of SNA/FS commands and server instructions by role.

Table 8-1. Base SNA/FS Commands and Instructions by Role

Command	**FP → EP**	**EP → FP**	**FP ↔ FP**
Base SNA/FS Commands			
omitted agent object	(note)	YES	YES
REPORT_FS_ACTION	YES	YES	YES
REPORTING_FS_ACTION	YES	YES	YES
TRANSFER_TO_REQUESTER	YES	(note)	YES
Base Server Instructions			
CREATE / LOAD_OR_REPLACE	YES	YES	YES
DECODE_ONLY	YES	YES	YES
DELETE	YES	—	YES
ENCODE_ONLY	—	—	—
FETCH	YES	YES	YES
SNA/FS Optional Sunset 1			
CREATE / LOAD	YES	—	—

Note: In normal operation, a focal point will never send an MU omitting an agent object to an entry point unless the entry point issues TRANSFER_TO_REQUESTER. However, an entry point is expected to accept one should it occur.

The following SNA/FS intentions are taken by management services:

- EXECUTING - Used when executable files are sent to entry points.

- STORING - Used when any files are sent to a focal point. Changes, for example, are not executed at focal points, but are stored for distribution.

The following SNA/FS exception actions are requested by management services:

- ABEND - If an SNA/FS server exception condition arises, the server is not required to return the storage files to their original condition.
- BACKOUT - If an SNA/FS server exception condition arises, the server *will* attempt to return the storage files to their original condition.

Bulk Data Commands and Replies:

To simplify processing and error reporting, each agent object, if present, will carry only one command or reply. These are described either by SNA/FS or management services.

The command sent by a focal point to another focal point or an entry point is determined by which request is made by the user. In addition, a REPORT_FS_ACTION command is used when a local user requests the entry point to send files to the focal point.

In both cases, a CP-MSU is carried in the agent object for management services commands.

A focal point is expected to provide the user with the capability of grouping requests into sequences with execution of subsequent requests conditioned on successful execution of prior requests. The user must be able to instruct the requesting node to begin execution of such groups of requests at a specific date and time.

Use of the SNA/FS Global File Name:

SNA/MS requires the use of *global* file names. That is, a file has a name that can be handled by each SNA/FS server in the network, and that can be recognized by each SNA/MS agent.

One of the important motivations for using SNA/FS is the requirement to minimize the implementation cost incurred by an SNA/MS focal point product to identify the files used by the wide variety of products in an SNA network. SNA/FS provides a global name for a file, consisting of a set of tokens. SNA/FS standardizes the number of tokens allowed (up to 10), the size of each token (up to 16 characters), total name length (65-n where n is the number of tokens), and the character set allowed for the token values (a limited set of character graphics displayable on most types of displays: Coded Graphics Character Set ID 01134-00500). In addition, SNA/FS architecture maintains a registration of values for the highest order token. For example, MCODE is the registered value for change files containing microcode. SNA/MS maintains a registration of the values of some of the other tokens, delegating authority in some cases to other administrative organizations. For example, the IBM machine type is the second token for microcode. As the result, a focal point is required to implement only one input panel, say, for a user to identify microcode files for a potentially wide variety of types of target entry points.

Definitions of token values by SNA/MS were made to satisfy two general requirements. The first requirement is that each file must be uniquely identified. The second requirement is to provide the user with some idea of the type and identity of the change when displaying the name, or to allow an application to process the token values. For example, the file name MCODE.9135.NA.PATCH.1234 indicates that the file contains a patch rather than an engineering change. This is needed both to uniquely identify the file, and also to provide useful information on display.

It is advantageous for the user to create (or have provided by product developers) change files that can be installed on a large number of entry points. This allows the fan-out feature of SNA/DS to be fully exploited, and reduces the user's effort. For example, files containing microcode can be designed to be applicable to many control units, whereas those containing customizing data are specific to individual control units.

Token values are assigned to each type of file used by change management. Refer to Appendix C, "SNA/FS File Names Defined by SNA/MS" on page C-1 for details about the tokens used by SNA/MS and associated rules defined for their use.

Use of SNA/FS Application Intervention Exit:

Products may implement their SNA/FS server to invoke an application exit based on file type. This allows closed protocol boundary implementations to perform agent processing before the first storage operation performed by the server. Some of the exception conditions recognized by change management are recognized in this context. The report codes are eventually returned from one agent to the other.

Use of SNA/FS Partial Name Processing: SNA/MS makes use of SNA/FS *partial name processing* for both retrieval and distribution. Partial name processing is used when the user wishes to specify only some of the file identification tokens (typically, the higher order ones). An example of this is when the latest version of a customizing data file is to be retrieved, but the user cannot remember the version number contained in the lowest identification token (and does not care what its value is).

On distribution, partial name processing may be needed to specify which file to destroy to make room for a new one, when entry point storage constraints arise. An important requirement addressed by the architecture is that destruction take place only when installation has been requested properly and pre-tests have been performed successfully. The complete identity of the file replaced is included in the report of successful installation.

Choice of SNA/DS Roles, Electives, and Options

Refer to *SNA/Distribution Services Reference, SC30-3098*, for details about the SNA/DS options.

A node participating in SNA/File Services must support SNA/DS format set 2 with only the following features required:

1. Role: End-only

   Focal points and entry points that do not implement the intermediate routing function can be configured to communicate through SNA/DS intermediate nodes and obtain the benefits of SNA/DS.

2. Use of correlator (to carry Agent Unit-of-Work Correlator)
3. Integrity: High
4. Protocol boundary: closed

   A product can choose to implement the SNA/DS layer solely for the purposes of network management. In doing so, an open protocol boundary would be useful if other applications will want to use it, but is not required.

5. Electives:

   - Receive-time enhancements

     Network management requires the capability to check more fields in the SNA/DS header than the minimum required (e.g. the destination DSU list).

   - Use of RGN

     RGN is needed to carry NETID.

In addition, the MU Operations function is recommended for attended nodes, because it provides the ability to list and purge distributions.

Because user names are not required for network management, certain optimizations are possible. In particular, a user directory and destination fan-out are *not* required.

Security

The following levels of security are available in LU 6.2 and SNA/DS to products implementing network management.

Refer to *SNA/Distribution Services Reference, SC30-3098*, and *SNA Transaction Programmer's Reference Manual for LU Type 6.2, GC30-3084*, for details.

1. LU 6.2 session-level security (LU-LU verification)
2. LU 6.2 conversation-level security verification
3. SNA/DS security, to ensure that the path traversed by a SNA/DS conversation is secure if so desired.
4. The SNA/DS origin name is provided to the SNA/FS server on the SNA/DS server protocol boundary, so that an intervention exit in the server can examine it.
5. Additional security to ensure that change management agents are trusted at a SNA/DS DSU may be provided by the node's operating system.

Network management data that is transported in an SNA/DS MU with the security service level must originate at a trusted DSU and always be transported in an MU specifying a security service level.

Identification of Resources and Application-Level Routing

Identifying SNA-Addressable Resources

SNA/Management Services identifies SNA-addressable resources with the SNA Address List (X'04') subvector. This subvector transports the network and local addresses by which resources are ordinarily identified within an SNA node. CPMS provides address-to-name and name-to-address translation for these addresses, so that a network operator viewing management services data such as Alerts, or entering management services commands, need only know resources by their names.

Identifying Resources That Are Not SNA-Addressable

Resources that are not SNA-addressable are identified in different ways in request records and in replies and unsolicited records that flow in the opposite direction. The two sections that follow detail these ways.

Requests

The Name List (X'06') subvector is the structure that identifies resources in requests. (This subvector also provides another capability, that of routing records to served network management applications; see "Routing a Request to a Specified Application" for a description of this capability.) The Associated Resource Name List (X'01') subfield is used in common operations services requests to identify the resource or set of resources to which a request applies. "Common Operations Services Commands, Replies, and Unsolicited Traffic" on page 10-144 contains a summary of the ways in which this subfield is used to identify resources in these requests.

Replies and Unsolicited Records

The Hierarchy/Resource List (X'05') subvector is the structure that identifies resources in replies and unsolicited records. The primary difference between it and the Name List (X'06') subvector is the provision for transporting code points that identify the *types* of resources involved, as well as their names.

Since the Alert (X'0000') major vector is the record that most fully utilizes the capabilities of the Hierarchy/Resource List subvector for identifying resources, the discussion of this subvector appears in the EP_ALERT section of this manual. See "Hierarchy Information in Alerts" on page 10-42 for this discussion.

Routing a Request to a Specified Application

The architecture for SNA/Management Services provides for the routing of management services traffic to specific applications running at a destination node. These applications are not the MS function set groups. Rather, they are network management applications served by an MS function set group. Since a request must first be routed to the serving function set group, and then to the served application, this type of routing is referred to as *second-level application routing.*

Second-level application routing utilizes the Destination Application Name (X'50') subfield within the Name List (X'06') subvector. This subfield transports a 1- to 8-character application identifier. When it receives a major vector containing this subfield, an MS function set group that supports routing of this type knows that the major vector is intended for an application that it serves. If it is aware of an application with the specified identifier, the MS function set group passes the major vector to it. Otherwise it rejects the major vector, sending sense data X'8018 0001' (application unknown).

Currently the only function set group providing second-level application routing is EP_COMMON_OPERATIONS_SERVICES.

Protocol Boundaries in the Management Services Model

The architectural model for management services presented in chapters 9—10 decomposes the overall network management capabilities in a node into a number of management services *function sets*. "Conventions Used in Describing Function Sets" summarizes the conventions used when these function sets are presented in chapters 9—10. "Protocol Boundaries Between MS Function Set Groups" on page 8-18 lists all of the lettered protocol boundaries over which data flows between management services function set groups.

Implementations are *not* required to match the architectural model of this book internally. An implementation can diverge from this formal model internally, but not in ways such that technical variations can be detected outside its node.

Conventions Used in Describing Function Sets

The function sets described in detail in Chapters 9—10 are based on the model previously introduced (see "Base and Optional Subsets of the Function Sets" on page 1-20). They state the architectural requirements for implementing the subset of PUMS that provides the function being described by the function set, and the subset of each of the other nodal components that assist PUMS.

Each function set has a corresponding function set group. A diagram of each function set group indicates the nodal components (or subsets of the nodal components) providing the function described by the function set, and the relationship of the subject function set group to other function set groups. Refer to Figure 8-7 on page 8-17.

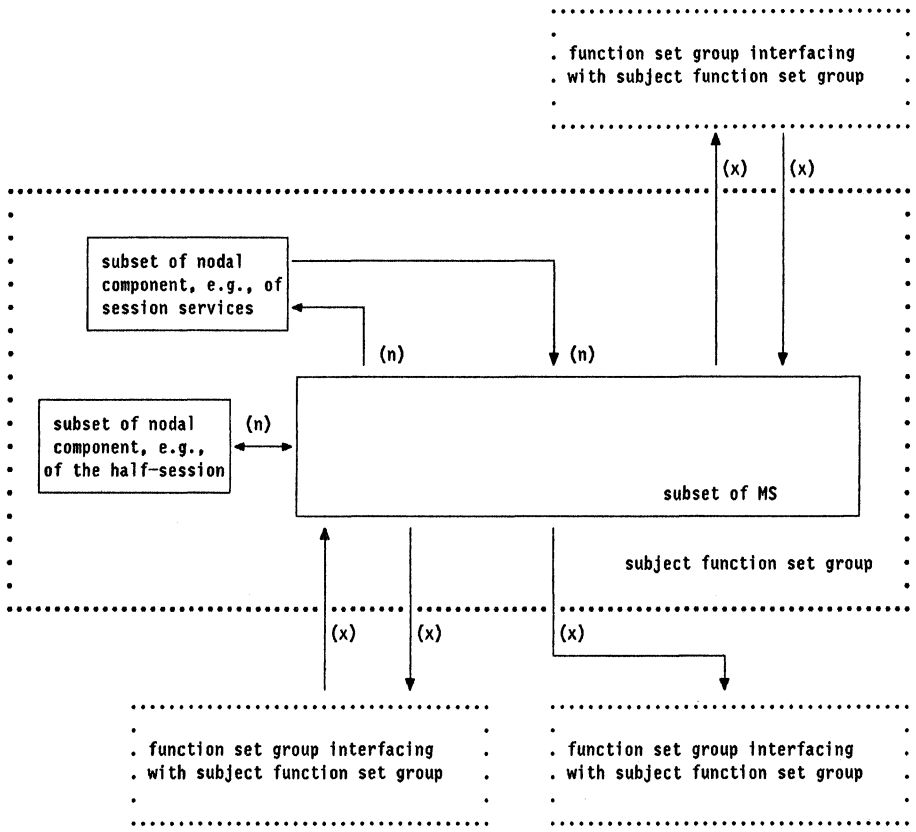


Figure 8-7. Conventions Used In Describing MS Function Set Groups

In each diagram, the function set group being described is bounded by a heavy dotted line (•••••). Each of the components responsible for providing function is shown within the boundary. This convention is not meant to show a physical composition, but only a functional composition. It is the subset of each component providing the function under discussion that the diagram depicts; thus a subset of PUMS is present within the subject function set group, and a different subset is present in other function set groups. The same is true for each component shown.

The protocol boundaries between MS components allocated to the subject function set group and MS components outside the subject function set group are shown and marked with letters for ease of reference in the text. They are indicated in Figure 8-7 by (x). The protocol boundaries between MS components allocated to the subject function set group and other components of the node allocated to the subject function set group are shown and marked with numbers for ease of reference in the text. They are indicated in Figure 8-7 by (n). The text describing the function will describe the protocol boundaries in detail.

Each function set group outside the subject function set group being described is bounded by a dotted line (.....). The functional composition of these function set groups is not shown.

Each function set falls into one of two categories, general or specialized. A general function set is a management services function set that provides a function used by multiple management services categories, such as transport of management services data. A specialized function set is a management services function set that processes data for a particular management services function. The table shown in Table 8-2 indicates each function set's category.

Table 8-2. Table of General and Specialized Function Sets for the PUMS Role

General Function Sets	Specialized Function Sets
SEND_DATA_SSCP_PU RECEIVE_REQUEST_SSCP_PU FILE_SERVICES_SUPPORT	EP_ALERT EP_RTM EP_QPI EP_CHANGE_MGMT EP_COMMON_OPERATIONS_SERVICES

Protocol Boundaries Between MS Function Set Groups

Protocol Boundary A - Send NMVT

Origin	EP_XXXX
Destination	SEND_DATA_SSCP_PU
Data Content	1. A pointer to the completed NMVT 2. For solicited data, the address of the control point that sent the request
Initiates	Process described in "Sending NMVTs" on page 9-14

Protocol Boundary B - Held Alert Processing

Origin	SEND_DATA_SSCP_PU
Destination	EP_ALERT (supporting optional subset 3 - Held Alert)
Data Content	1. A pointer to a completely constructed NMVT If the pointer is zero, EP_ALERT is being requested to process an existing held Alert. If the pointer is non-zero, it contains a pointer to an NMVT to be queued.
Initiates	Process described in "Sending Held Alerts" on page 10-15

Protocol Boundary C - Delayed Alert Processing

Origin	SEND_DATA_SSCP_PU or EP_ALERT
Destination	EP_ALERT (implementing optional subset 2 - Delayed Alert) or SEND_DATA_SSCP_PU
Request Data Content	1. A pointer to a completely constructed NMVT
Reply Data Content	1. An indicator whether the NMVT can be sent at this time If the indicator is zero, the calling process can send the NMVT at this time. If the indicator is non-zero, the NMVT cannot be sent at this time and is being processed by this optional subset.
Initiates	Process described in "Processing Delayed Alerts" on page 10-12

Protocol Boundary D - NMVT Received

Origin	RECEIVE_REQUEST_SSCP_PU
Destination	EP_XXXX
Data Content	1. A pointer to the NMVT 2. The address of the control point from which the NMVT was received
Initiates	Passing data across this protocol boundary causes one of the following processes to be started, determined by the key of the major vector in the NMVT referred to by the pointer in item 1. • "Receiving RTM Requests" on page 10-55 • "Receiving QPI requests" on page 10-68 • "Receiving Common Operations Services Requests" on page 10-142

Protocol Boundary E - Send NMVT Response

Origin	EP_XXXX
Destination	RECEIVE_REQUEST_SSCP_PU
Data Content	The following items, used to construct and send an NMVT response RU: 1. The address of the control point which is to be sent a +RSP or -RSP 2. Either the 8-digit hexadecimal sense data that is to be sent on a -RSP, or the value X'0000 0000', indicating that a +RSP is to be sent
Initiates	Process described in "Sending NMVT Responses" on page 9-19

Protocol Boundary F - Send MS Bulk Data

Origin	EP_XXXX
Destination	FILE_SERVICES_SUPPORT
Data Content	Information defining the FS request, consisting of these elements: 1. CP-MSU, to be placed in the agent object 2. Agent unit-of-work correlator 3. Server object parameters 4. List of target locations (NETID and LUNAME for each) 5. SNA/DS priority and security level requested
Initiates	"Sending Requests and Data" on page 9-6

Protocol Boundary G - MS Bulk Data Received

Origin	FILE_SERVICES_SUPPORT
Destination	EP_XXXX
Data Content	Information defining the FS report, consisting of these elements: 1. CP-MSU 2. Agent unit-of-work correlator 3. Server object parameters 4. Source location (NETID and LUNAME)
Initiates	Appropriate specialized management services function set process for receiving data

Role Requirements for Management Services Components

This section introduces the next two chapters by discussing the conventions used and function sets described in those chapters, and it specifies for the implementer, the role available for a management services component. The reader of this section should be familiar with the material in "Implementation Choices" on page 1-20.

This section describes the function sets required for the role of PUMS in a type 2.0 node, the function sets optionally available for this role, and the dependence of the function sets upon one another.

Physical Unit Management Services (PUMS) in a Type 2.0 Node

Introductory material on roles can be found in "Role Requirements" on page 1-21 and Figure 1-6 on page 1-22.

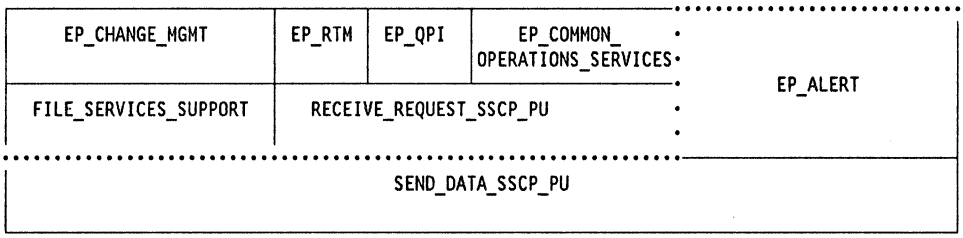


Figure 8-8. Function Sets for the PUMS Type 2.0 Role

All implementations of type 2.0 nodes[1] must provide the management services described in the base subset of the following:

- "SEND_DATA_SSCP_PU Function Set" on page 9-11
- "EP_ALERT Function Set" on page 10-3

The following function sets are available as options to implementations of type 2.0 nodes. Refer to Figure 8-8 for the relationship of these function sets to one another.

- "RECEIVE_REQUEST_SSCP_PU Function Set" on page 9-16
- "FILE_SERVICES_SUPPORT Function Set" on page 9-3
- "EP_CHANGE_MGMT Function Set" on page 10-71
- "EP_RTM Function Set" on page 10-51
- "EP_QPI Function Set" on page 10-65
- "EP_COMMON_OPERATIONS_SERVICES Function Set" on page 10-138

[1] This role also applies to boundary-function-attached type 2.1 nodes.

Chapter 9. General Management Services Function Sets

Part II

FILE_SERVICES_SUPPORT Function Set

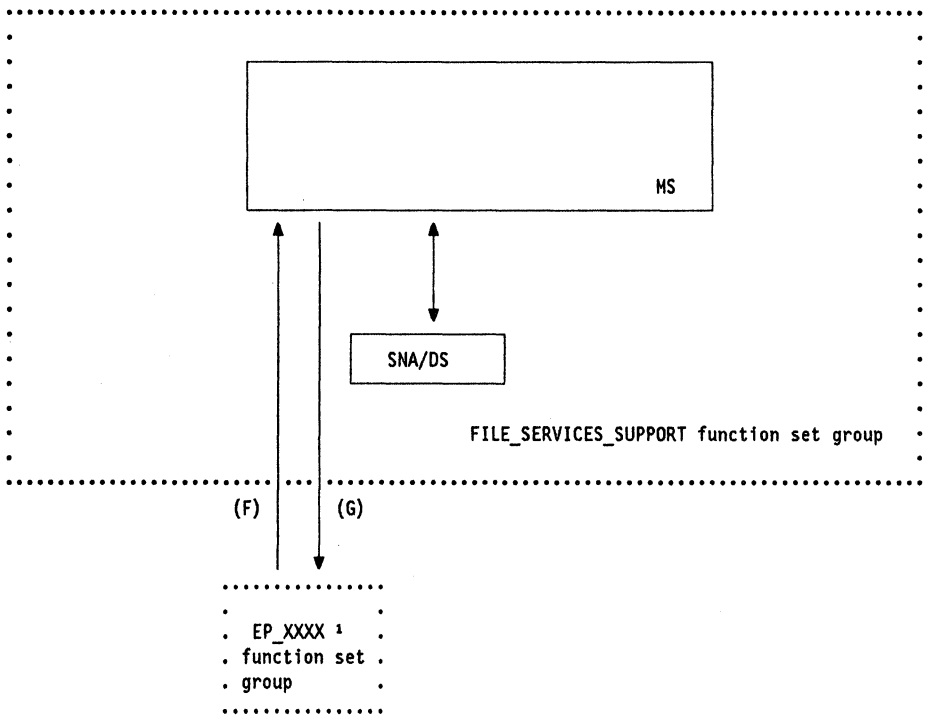


Figure 9-1. FILE_SERVICES_SUPPORT Function Set Group

The FILE_SERVICES_SUPPORT function set provides the capability to route MS commands, reports, and bulk data between MS function set groups in different nodes.

Refer to Figure 9-1 throughout the discussion of the FILE_SERVICES_SUPPORT function set.

For detailed information on the prerequisite architecture components, refer to *SNA/File Services Reference, SC31-6807*, and *SNA/Distribution Services Reference, SC30-3098*.

Protocol Boundaries with Components Outside FILE_SERVICES_SUPPORT

- Input
  - Internal MS protocol boundary F (Request File Services Support)

    This function set group receives requests from the EP_XXXX function set group to route bulk MS data and requests to other nodes. The details of this protocol boundary are described in "Protocol Boundary F - Send MS Bulk Data" on page 8-20.
- Output

- Internal MS protocol boundary G (File Services Report Support)

  This function set group reports to the EP_XXXX function set group about the status of previous requests to route bulk MS data and requests to other nodes, or about the arrival of unsolicited bulk data or requests. The details of this protocol boundary are described in "Protocol Boundary G - MS Bulk Data Received" on page 8-20.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

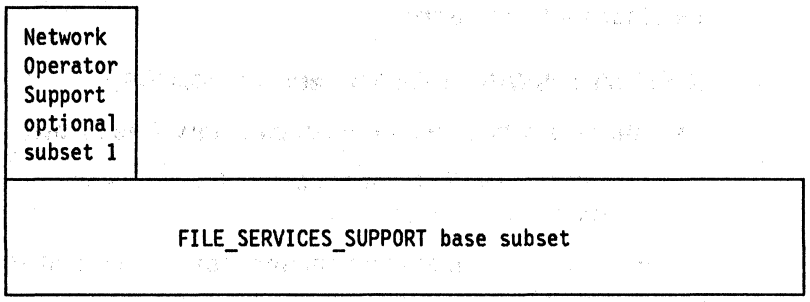


Figure 9-2. Base and Optional Subsets of FILE_SERVICES_SUPPORT Function Set

FILE_SERVICES_SUPPORT Base Subset

Functions Provided

The FILE_SERVICES_SUPPORT base subset provides the ability to route management services requests and bulk data between function set groups located at separate LUs. This routing is accomplished over LU-LU sessions used by SNA/Distribution Services (SNA/DS).

Formats Supported

This base subset provides:

- The capability to send and receive CP-MSUs, SNA/FS agent objects, and SNA/FS files (bulk data) over LU-LU sessions.
- Support for both processing and building SNA/FS agent objects, containing one of the following:
  - TRANSFER_TO_REQUESTER command
  - REPORT_FS_ACTION command

where the SNA/FS server instruction in the server object is one of the following:

- CREATE/LOAD, or
- CREATE/LOAD_OR_REPLACE

- REPORTING_FS_ACTION report

and support for processing, but not building:

- REPORT_FS_ACTION command

  where the SNA/FS server instruction in the server object is:

  - DELETE

Implementation Requirements

The requirements for implementing the FILE_SERVICES_SUPPORT base subset are described by a model consisting of a subset of SNA/Distribution Services and SNA/Management Services.

A Subset of SNA/Distribution Services (SNA/DS):

- Interacts with MS via protocol boundary 7 as follows:
  - Notifies MS that a management services MU and bulk data have arrived on a conversation
  - Receives a management services MU and bulk data from another SNA/DS node
  - Sends an MU and bulk data to the specified LU on a conversation when requested by MS

A Subset of SNA/Management Services:

- Provides the following processes:
  - Sending requests and data
    - Receives CP-MSUs and SNA/FS parameters from specialized management services function set groups on protocol boundary F
    - Sends SNA/FS agent objects and bulk data to other nodes, sending them on a conversation via SNA/DS
    - Passes back SNA/DS exception conditions if the agent object and bulk data could not be sent
  - Receiving requests and data
    - Receives CP-MSUs and bulk data from other nodes on an SNA/DS conversation and routes the data to the appropriate XXXX_NETOP or EP_XXXX specialized function set group on protocol boundary G
    - Receives SNA/FS agent objects and bulk data from other nodes on an SNA/DS conversation and processes them
  - Processing SNA/FS Agent Objects

— Parses a SNA/FS agent object and honors the request contained within it, which is of one of the following kinds:

  - To send bulk data to another node, or to build and send an SNA/FS report if there are exceptions in attempting to do so, or
  - To build and send an SNA/FS report indicating either success or failure in attempting to store a file at this node.

Sending Requests and Data:

This process is started by one of the following:

- A specialized management services function set group that has CP-MSU and bulk data destined for a remote LU. The process is started via internal MS protocol boundary F and is passed a CP-MSU, an agent unit-of-work correlator, SNA/FS server object parameters, and a network-qualified LU name.
- The process described in "Processing SNA/FS Agent Objects" on page 9-7.
- The network operator, if the Network Operator Support optional subset is implemented.

After being started, this process issues SEND_DISTRIBUTION to pass the agent object, along with SNA/FS parameters, to SNA/Distribution Services.

Note: Refer to the SNA/DS and SNA/FS references listed in "Prerequisite Publications" on page vi for a complete description of the SNA/DS verbs and included SNA/FS parameters.

SEND_DISTRIBUTION is prepared as follows:

- DISTRIBUTION_ID specifies the network-qualified LU name of this LU
- AGENT_CORREL specifies the correlation value
- DESTINATION specifies the network-qualified destination LU name passed via protocol boundary F
- DEST_AGENT specifies SNA/Management Services (X'23F0F0F0')
- AGENT_OBJECT contains either the CP-MSU or the SNA/FS agent object
- SERVICE_PARMS specifies
  - priority=DATA_12 or lower
  - protection=YES
  - capacity=16 MEG
  - security=YES OR NO
  - accept_delay=INDEFINITE
- REPORTING_REQUESTED specifies
  - exception_report_req=YES
- INTEGRITY specifies HIGH

- REPORT-TO_DSU specifies the network-qualified LU name of the focal point that initiated this unit of work
- SERVER specifies SNA/File Services (X'24F0F0F0')
- SPECIFIC_SERVER_INFO specifies the SNA/FS parameters

The return code from the execution of the verb along with any SNA/DS exception information is passed back to the process that started this process.

This process then terminates.

Receiving Requests and Data:

This process is started by SNA/DS in this node when it receives data for SNA/MS (as specified by the agent TP name in the MU). After being started, it issues RECEIVE_DISTRIBUTION to receive the data sent to it by a remote LU The following returned values are used by this process:

- DISTRIBUTION_ID specifies the network-qualified LU name of the LU that the MU was received from
- AGENT_CORREL specifies the Agent Unit-of-work Correlator (X'1549') GDS variable sent by the remote LU
- ORIGIN specifies the network-qualified origin LU name to be passed via protocol boundary G
- AGENT_OBJECT (if specified) specifies one of the following:
  - A CP-MSU (X'1212') GDS variable carrying a major vector, or
  - An FS command
- REPORTING_REQUESTED specifies
  - exception_report_req=YES
- REPORT-TO_DSU specifies the network-qualified LU name of the originator of the unit of work
- DISTRIBUTION_TIME specifies the time at which the distribution originated
- SPECIFIC_SERVER_INFO specifies the SNA/FS parameters
- RETURN_CODE specifies results of RECEIVE_DISTRIBUTION verb execution.

If the agent object contains a CP-MSU, this process then passes it along with other required parameters to the appropriate specialized management services function set via internal MS protocol boundary G and terminates.

Otherwise (if the agent object does not contain a CP-MSU), this process starts the process described in "Processing SNA/FS Agent Objects."

Processing SNA/FS Agent Objects:

This process is started when the process described in "Receiving Requests and Data" receives an FS command. After being started, this process parses the agent object.

Refer to Table 9-1 on page 9-8.

Table 9-1. Choosing Agent Object Contents and Server Parameters from the Command Received		
SNA/FS command received	**SNA/FS agent object contents to send**	**SNA/FS server instructions to send (E=encoder D=decoder S=source T=target)**
TRANSFER_ TO_ REQUESTER	NONE (AGENT OBJECT IS OMITTED)	S: (FETCH, ABEND, ONLY_ IF_ EXCEPTIONS) T: (CREATE/LOAD_ OR_ REPLACE, STORING, ABEND, DETAILED)
REPORT_ FS_ ACTION	REPORTING_ FS_ ACTION	E: (ENCODE_ ONLY, ABEND, ONLY_ IF_ EXCEPTIONS) D: (DECODE_ ONLY, ABEND, DETAILED)

Notes:

1. In the case of TRANSFER_TO_REQUESTER, the file name is a Fetched File Name if SNA/FS action is successful, a "To-Be-Fetched File Name if it is not.
2. In the REPORTING_FS_ACTION report, the file name is a Stored Name or a Deleted Name if SNA/FS action is successful, a To-Be-Stored Name or To-Be-Deleted Name if it is not.

If the agent object contains a TRANSFER_TO_REQUESTER command, this process uses the specified FS parameters required to fetch the data object, and issues SEND_DISTRIBUTION to transfer the file to the requesting focal point, using the same parameters as described in "Sending Requests and Data" on page 9-6, but omitting AGENT_OBJECT.

If the agent object contains a REPORT_FS_ACTION command, this process encodes FS parameters required to report on the status of the stored data object, builds an FS agent object containing a REPORTING_FS_ACTION command, and then issues SEND_DISTRIBUTION.

If this process is unable to parse the agent object, or in any other exception conditions, it builds an agent object containing an SNA Condition Report (X'1532') GDS variable and issues SEND_DISTRIBUTION.

This process then terminates.

FILE_SERVICES_SUPPORT Optional Subset 1 (Network Operator Support)

Functions Provided

Optional Subset 1 (Network Operator Support) provides the capability to interact with the network operator at the node to receive request verbs and pass back reply verbs. Refer to Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 for a detailed description of these verbs.

Verbs Supported

This optional subset provides support for receiving and processing the following verbs from the network operator:

- SEND
- RETRIEVE

and issuing the following verbs to the network operator:

- REPLY_TO_SEND
- REPLY_TO_RETRIEVE
- NOTIFICATION_OF_ARRIVAL

Implementation Requirements

The additional requirements for implementing this optional subset are described by a model consisting of the following:

A Subset of SNA/Management Services:

- Provides the following processes:
  - Sending requests and data
    - Receives SNA/FS parameters from the network operator, and sends them after the SNA/FS agent object is built to other nodes on a conversation via SNA/DS
    - Passes back SNA/DS exception conditions if the agent object and bulk data could not be sent
  - Receiving data
    - Receives SNA/FS agent object reports and passes them to the network operator after converting them to the reply verb format.
  - Processing SNA/FS Agent Objects
    - Builds a SNA/FS agent object from a network operator verb

Sending Requests and Data:

The process described in "Sending Requests and Data" on page 9-6 is enhanced to support being started by:

- The network operator.

Then, the process described in "Processing SNA/FS Agent Objects" on page 9-7 (enhanced as described below) is started to build the required SNA/FS agent object from the verb. Otherwise, processing is the same.

Receiving Requests:

The process described in "Receiving Requests and Data" on page 9-7 is enhanced to convert SNA/FS reports and notifications of the arrival of files into reply verbs. The appropriate reply verb is chosen in an obvious manner from the request with which it is correlated using the Agent Unit-of-Work correlator:

- An outstanding SEND request will result in receipt of a REPORTING_FS_ACTION report, and the REPLY_TO_SEND reply verb is built.
- An outstanding RETRIEVE request will result in receipt of a file but no agent object, and the REPLY_TO_RETRIEVE reply verb is built.

This process then passes the reply verb to the network operator and terminates.

Processing SNA/FS Agent Objects:

The process described in "Processing SNA/FS Agent Objects" on page 9-7 is enhanced:

- To build SNA/FS agent objects from the verb issued by the network operator.
- To pass REPORTING_FS_ACTION reports back to the process described in "Receiving Requests and Data" on page 9-7.

This process uses the following table to determine which kind of agent object to build, and which server instructions to include in the SNA/FS parameters. After building this data, it returns control to the process that started it and terminates.

Table 9-2. Choosing Command and Server Instructions from the Verb

Verb issued by network operator	**SNA/FS Command**	**SNA/FS Server Instructions (E=encoder D=decoder S=source T=target)**
SEND DESTRUCTION (ALLOWED)	REPORT_ FS_ ACTION	S: (FETCH, ABEND, ONLY_ IF_ EXCEPTIONS) T: (CREATE/LOAD_ OR_ REPLACE, EXECUTING, ABEND, DETAILED) FOR ENTRY POINT TARGETS T: (CREATE/LOAD_ OR_ REPLACE, STORING, ABEND, DETAILED) FOR FOCAL POINT TARGETS
SEND DESTRUCTION (NO)	REPORT_ FS_ ACTION	S: (FETCH, ABEND, ONLY_ IF_ EXCEPTIONS) T: (CREATE/LOAD, EXECUTING, BACKOUT, DETAILED) FOR ENTRY POINT TARGETS T: (CREATE/LOAD, STORING, BACKOUT, DETAILED) FOR FOCAL POINT TARGETS
RETRIEVE	TRANSFER_ TO_ REQUESTER	E: (ENCODE_ ONLY, ABEND, ONLY_ IF_ EXCEPTIONS) D: (DECODE_ ONLY, ABEND, DETAILED) S: (FETCH, ABEND, ONLY_ IF_ EXCEPTIONS) T: (CREATE/LOAD_ OR_ REPLACE, STORING, ABEND, DETAILED)

Notes:

1. For REPORT_FS_ACTION,

   TARGET_AGENT_REPORTING_ACTION = DETAILED.

2. For TRANSFER_TO_REQUESTER,

   SOURCE_AGENT_REPORTING_ACTION = ONLY_IF_EXCEPTIONS and
   TARGET_AGENT_REPORTING_ACTION = ONLY_IF_EXCEPTIONS.

3. The REPORT_TO location is omitted, since the REPORT_TO party is the requester, or target agent.

SEND_DATA_SSCP_PU Function Set

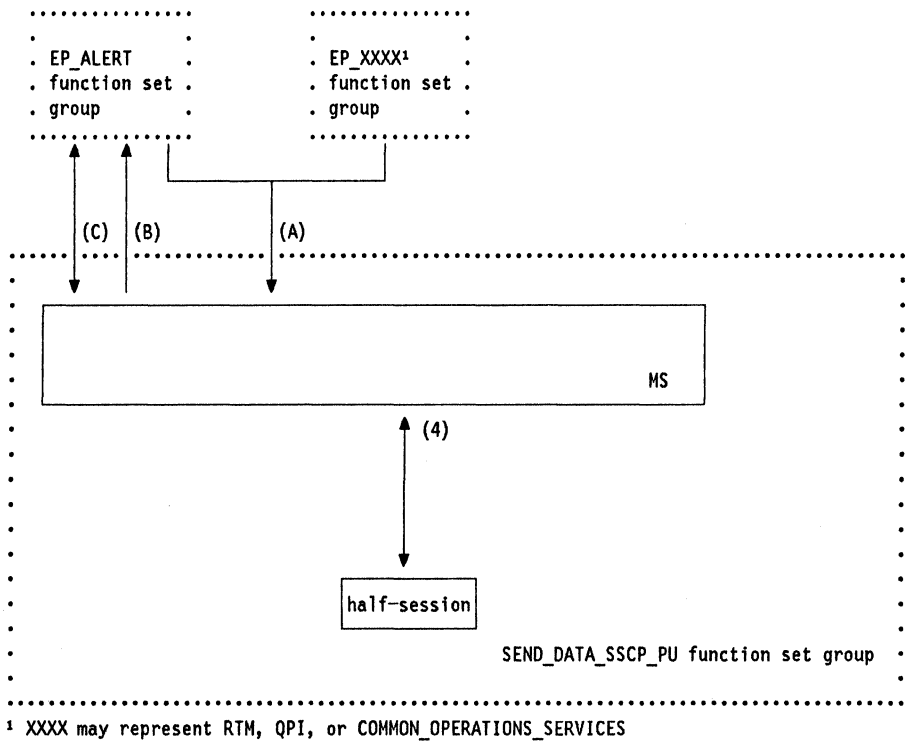


Figure 9-3. SEND_DATA_SSCP_PU Function Set Group

The SEND_DATA_SSCP_PU function set is a base function set for all PUMS implementations. It includes all support necessary for sending NMVT RUs to CPMS on an SSCP-PU session.

Refer to Figure 9-3 throughout the discussion of the SEND_DATA_SSCP_PU function set.

Protocol Boundaries with Components Outside SEND_DATA_SSCP_PU

- Input:
  - Internal MS protocol boundary A (Send NMVT)

    The SEND_DATA_SSCP_PU function set group receives requests from the following function set groups to send an NMVT RU on the SSCP-PU session.

    - EP_ALERT
    - EP_RTM
    - EP_QPI
    - EP_COMMON_OPERATIONS_SERVICES

    The input data consists of a complete NMVT built by the requesting function set group. The details of this protocol boundary are described in "Protocol Boundary A - Send NMVT" on page 8-18.

- Internal MS protocol boundary C (Delayed Alert Processing)

  If EP_ALERT optional subset 2 (Delayed Alert) is implemented, the SEND_DATA_SSCP_PU function set group requests the EP_ALERT function set group to process an NMVT to determine whether it can be transmitted at this time, classified as a delayed Alert to be transmitted at a later time, or logged if it cannot be transmitted at all. After issuing such a request, this function set group will receive input from the EP_ALERT function set group indicating whether the NMVT can be transmitted. The details of this protocol boundary are described in "Protocol Boundary C - Delayed Alert Processing" on page 8-18.

- Output:
  - Internal MS protocol boundary B (Held Alert Processing)

    If EP_ALERT optional subset 3 (Held Alert) is implemented, the SEND_DATA_SSCP_PU function set group requests the EP_ALERT function set group to do the following:

    - Queue an Alert NMVT to be sent at a later time
    - Log a non-Alert NMVT that cannot be sent at this time, or an Alert NMVT that cannot be queued
    - Remove an Alert from the held Alert queue and pass it to the SEND_DATA_SSCP_PU function set group to be sent on the SSCP-PU session.

    The details of this protocol boundary are described in "Protocol Boundary B - Held Alert Processing" on page 8-18.

  - Internal MS protocol boundary C (Delayed Alert Processing)

    If EP_ALERT optional subset 2 (Delayed Alert) is implemented, the SEND_DATA_SSCP_PU function set group requests the EP_ALERT function set group to process an NMVT to determine whether it can be transmitted at this time, classified as a delayed Alert to be transmitted at a later time, or logged if it cannot be transmitted at all. The details of this protocol boundary are described in "Protocol Boundary C - Delayed Alert Processing" on page 8-18.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

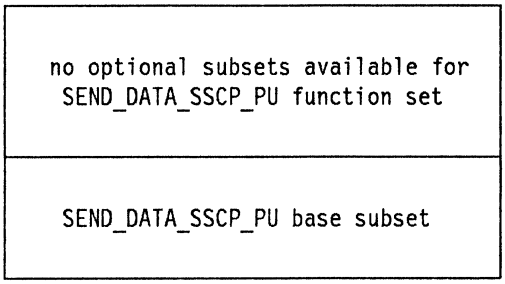


Figure 9-4. Base and Optional Subsets of SEND_DATA_SSCP_PU Function Set

SEND_DATA_SSCP_PU Base Subset

Functions Provided

PUMS communicates with CPMS on an SSCP-PU session by means of the NMVT. A brief description and example flow for each of these RUs is presented in Chapters 3, 4, 5, and 7. SEND_DATA_SSCP_PU provides the following support for sending the NMVT RU:

- Causes an NMVT to be sent on the SSCP-PU session
- Deals with the +RSP to an NMVT from CPMS
- Assists the EP_ALERT function set group in the processing of delayed and held Alerts

Formats Supported

This function set group supports the sending of NMVTs, and the receipt of responses to NMVTs.

Implementation Requirements

The requirements for implementing the SEND_DATA_SSCP_PU base subset are described by a model consisting of a subset of PU configuration services, the half-session and PU management services.

A Subset of the Half-Session:

- Interacts with PUMS via PUMS protocol boundary 4 as follows:
  - Notifies PUMS that an NMVT response RU has arrived on an SSCP-PU session
  - Receives an NMVT response RU from an SSCP-PU session when requested by PUMS
  - Sends an NMVT RU on a specified SSCP-PU session when requested by PUMS

A Subset of PU Management Services:

- Provides the following processes:
  - Sending NMVTs
    - Sends an NMVT on the SSCP-PU session when requested by other function set groups via protocol boundary A
    - Assists the EP_ALERT function set group in the processing of held and delayed Alerts
  - Receiving NMVT responses
    - Receives an NMVT response when notified by the half-session
    - Assists the EP_ALERT function set group in the processing of held Alerts

Sending NMVTs:

This process receives NMVTs from other function set groups and causes them to be sent to CPMS.

This process is started by the noted processes in the following function set groups:

- EP_ALERT ("Sending Alerts" on page 10-8)
- EP_RTM ("Sending RTM Data" on page 10-59)
- EP_QPI ("Sending QPI data" on page 10-69)
- EP_COMMON_OPERATIONS_SERVICES ("Sending Common Operations Services Data" on page 10-143)

When one of these function set groups starts this process, it passes a complete NMVT to it via protocol boundary A. Refer to "Protocol Boundary A - Send NMVT" on page 8-18 for details. After it is started, the process proceeds as follows:

It checks for the existence of the delayed Alert control block, and, if it is present, starts the process described in "Processing Delayed Alerts" on page 10-12, passing it a pointer to the NMVT to be transmitted via internal MS protocol boundary C item 1. See "Protocol Boundary C - Delayed Alert Processing" on page 8-18. The delayed Alert control block is present if optional subset 1 (Delayed Alert) of the EP_ALERT function set is implemented. If the delayed Alert control block is empty, and if the current NMVT is not itself a delayed Alert, then this process can send this NMVT to CPMS. This is indicated by returned item 1 from the Delayed Alert process. See "Protocol Boundary C - Delayed Alert Processing" on page 8-18. If the returned value is non-zero, the NMVT cannot be transmitted and this process ends.

If the PRID value is non-zero (i.e., if the NMVT was solicited) this process requests the half-session to send the NMVT that it has constructed to the control point address passed to this process in item 2, i.e., the control point that solicited it.

If the PRID value is X'000', (indicating unsolicited data), then this process requests the half-session to send the NMVT it has constructed to CPMS.

Figure 10-4 on page 10-15 shows the structure of the held Alert control block. If the half-session was unable to send an NMVT, and if the held Alert control block is present (if optional subset 3 of the EP_ALERT function set is implemented), this process starts the process described in "Sending Held Alerts" on page 10-15 via internal MS protocol boundary B. A pointer to the completed NMVT is passed in item 1. See "Protocol Boundary B - Held Alert Processing" on page 8-18.

After this process has sent each NMVT it has received, passed it (when possible) to the Delayed Alert or Send Held Alert process, to be saved as a delayed or held Alert, or had it logged locally, it terminates.

Receiving NMVT Responses:

This process receives responses to an NMVT from a control point and (if optional subset 3 of the EP_ALERT function set is implemented) starts the process described in "Sending Held Alerts" on page 10-15 if any Alerts have been held.

When a response to an NMVT arrives on the SSCP-PU session, the half-session starts this process which receives the response.

Figure 10-4 on page 10-15 shows the structure of the held Alert control block. If the held Alert control block is present (if optional subset 3 of the EP_ALERT function set is implemented), and if HEAD_POINTER is non-zero, this process starts the process described in "Sending Held Alerts" on page 10-15 via internal MS protocol boundary B. Zero is specified in item 1. See "Protocol Boundary B - Held Alert Processing" on page 8-18.

This process then terminates.

RECEIVE_REQUEST_SSCP_PU Function Set

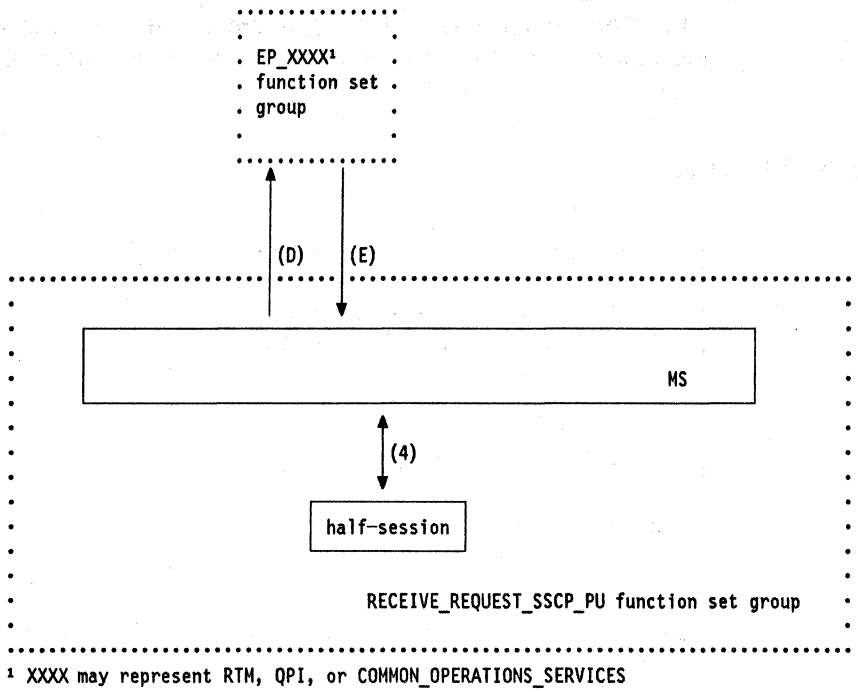


Figure 9-5. RECEIVE_REQUEST_SSCP_PU Function Set Group

The RECEIVE_REQUEST_SSCP_PU function set provides the basic capability to receive NMVTs and send a response. It is a prerequisite for any function sets supporting an NMVT request.

Refer to Figure 9-5 throughout the discussion of the RECEIVE_REQUEST_SSCP_PU function set.

Protocol Boundaries with Components Outside RECEIVE_REQUEST_SSCP_PU

- Input:
  - Internal MS protocol boundary E (Send NMVT Response)

    The RECEIVE_REQUEST_SSCP_PU function set group receives requests from other function set groups to send an NMVT response RU on an SSCP-PU session with a specified CP. The input data consists of the CP address and sense data. The details of this protocol boundary are described in "Protocol Boundary E - Send NMVT Response" on page 8-19.

- Output:
  - Internal MS protocol boundary D (NMVT Received)

    The RECEIVE_REQUEST_SSCP_PU function set group requests other function set groups to process incoming NMVTs by passing them an NMVT and the address of the CP from which it was received. The details of this pro-

tocol boundary are described in "Protocol Boundary D - NMVT Received" on page 8-19.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

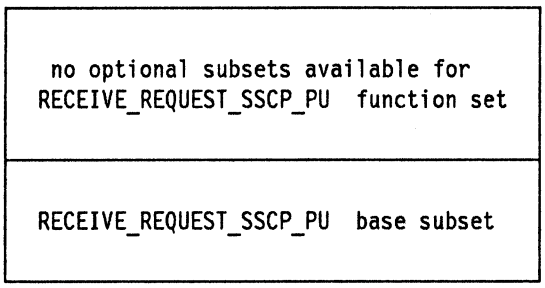


Figure 9-6. Base and Optional Subsets of RECEIVE_REQUEST_SSCP_PU Function Set

RECEIVE_REQUEST_SSCP_PU Base Subset

Functions Provided

RECEIVE_REQUEST_SSCP_PU base subset provides the basic capability to receive NMVTs, perform basic parsing of the NMVT, and pass the NMVT to the appropriate function set group or send a negative response if errors are detected while parsing. It is a prerequisite for any function set supporting an NMVT request.

Formats Supported

The RECEIVE_REQUEST_SSCP_PU function set group provides the ability to receive NMVTs and do the processing detailed below. It requires other function set groups to complete NMVT processing.

Implementation Requirements

The requirements for implementing the RECEIVE_REQUEST_SSCP_PU base subset are described by a model consisting of subsets of the half-session and PU management services.

A Subset of the Half-Session:

- Interacts with PUMS via PUMS protocol boundary 4
  - Notifies PUMS that an NMVT RU has arrived on an SSCP-PU session
  - Receives an NMVT RU from an SSCP-PU session when requested by PUMS

- Sends an NMVT response RU on an SSCP-PU session when requested by PUMS

A Subset of PU Management Services:

- Provides the following processes:
  - Receiving NMVTs
    - When notified by the half-session, receives an NMVT, performs preliminary parsing, and does the following:
      - If no errors were detected in parsing, passes the NMVT to the appropriate function set group (via protocol boundary D)
      - If errors were detected in parsing, starts the process described in "Sending NMVT Responses" on page 9-19 (via protocol boundary E) to send a negative response
  - Sending NMVT responses
    - Constructs an NMVT response RU and sends it on an SSCP-PU session when requested by other function set groups (via protocol boundary E)

Receiving NMVTs:

When a request NMVT arrives on the SSCP-PU session, the following sequence of events takes place (refer to "Parsing of NMVTs" on page 10-152 for details of common NMVT parsing):

1. The half-session manager examines the NS header to determine where within the PU to route the request.
2. Since the header is X'41038D', indicating that the RU is an NMVT, the half-session manager does one of the following:
   - If PUMS in the node does not support NMVTs, the half-session manager sends a -RSP (1003 0001) to indicate that NMVTs are not supported.
   - If PUMS is not prepared to process an NMVT, the PU resources manager sends a -RSP (0815 0003) to indicate that another management services request is currently being processed and the request cannot be accepted at this time.
   - If PUMS in the node supports NMVTs, and if it is currently prepared to process one, the half-session manager starts this process.
3. Having been started by the half-session manager, this process receives the NMVT. It is passed the length of the NMVT, as well as an indication of what control point sent the NMVT. Optionally, this process locates the major vector length in the NMVT. (As indicated in Table 10-37 on page 10-152, checking for an invalid major vector length is an option for an implementation of PUMS. The PUMS model presented here will include all optional checks, for completeness.) If the major vector length is incompatible with the RU length in the transmission header, this process starts the process described in "Sending NMVT Responses" on page 9-19 (RECEIVE_REQUEST_SSCP_PU function set), passing to it the sense data value

X'086F 0001'. That process sends a -RSP containing this sense data value to CPMS.

4. If the major vector length is compatible with the RU length in the transmission header, this process locates the major vector key in the NMVT.

   - If the major vector key is X'8080', and the EP_RTM function set has been implemented, then this process starts the process described in "Receiving RTM Requests" on page 10-55, passing the NMVT pointer and the address of the sending control point to it.
   - If the major vector key is X'8090', and the EP_QPI function set has been implemented, then this process starts the process described in "Receiving QPI requests" on page 10-68, passing the NMVT pointer and the address of the sending control point to it.
   - If the major vector key is X'8061', X'8062', X'8063', or X'8064', and the EP_COMMON_OPERATIONS_SERVICES function set has been implemented, then this process starts the process described in "Receiving Common Operations Services Requests" on page 10-142, passing the NMVT pointer and the address of the sending control point to it.

   See "Protocol Boundary D - NMVT Received" on page 8-19 for details of the protocol boundary used in the preceding cases.

   - If the major vector key specifies a function not supported by this instance of PUMS, then this process starts the process described in "Sending NMVT Responses," passing to it the sense data value X'080C 0005' and the address of the sending control point. That process will send a -RSP containing this value to CPMS in the control point that sent the request.

5. The process started by this process executes, then it returns control to this process, which terminates.

Sending NMVT Responses:

This process is started by the noted processes of the following function set groups:

- RECEIVE_REQUEST_SSCP_PU ("Receiving NMVTs" on page 9-18)
- EP_RTM ("Receiving RTM Requests" on page 10-55)
- EP_QPI ("Receiving QPI requests" on page 10-68)
- EP_COMMON_OPERATIONS_SERVICES ("Receiving Common Operations Services Requests" on page 10-142)

See "Protocol Boundary E - Send NMVT Response" on page 8-19 for a description of the protocol boundary by which this process was started.

After it has sent a response, this process returns control to the process that started it.

Chapter 10. Specialized Management Services Function Sets for Entry Points

EP_ALERT Function Set

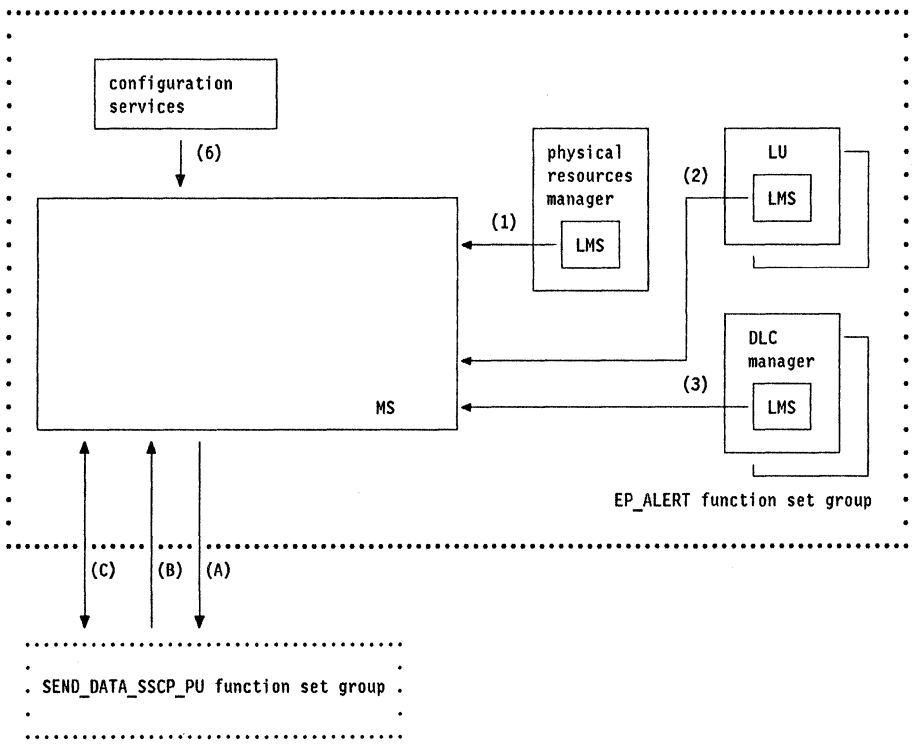


Figure 10-1. EP_ALERT Function Set Group

The EP_ALERT function set is responsible for sending Alerts.

Refer to Figure 10-1 throughout the discussion of the EP_ALERT function set.

Protocol Boundaries with Components Outside EP_ALERT

- Input:
  - Internal MS protocol boundary B (Held Alert Processing)

    If optional subset 3 (Held Alert) is implemented, the EP_ALERT function set group receives requests from the SEND_DATA_SSCP_PU function set group to do the following:

    - Queue an Alert NMVT to be sent at a later time
    - Log a non-Alert NMVT, or an Alert NMVT that cannot be queued
    - Remove an Alert from the held Alert queue and pass it to the SEND_DATA_SSCP_PU function set group to be sent on the SSCP-PU session

    The details of this protocol boundary are described in "Protocol Boundary B - Held Alert Processing" on page 8-18.

  - Internal MS protocol boundary C (Delayed Alert Processing)

If optional subset 2 (Delayed Alert) is implemented, the EP_ALERT function set group receives requests from the SEND_DATA_SSCP_PU function set group to process an NMVT to determine whether it can be transmitted at this time, classified as a delayed Alert to be transmitted at a later time, or logged if it cannot be transmitted at all. The details of this protocol boundary are described in "Protocol Boundary C - Delayed Alert Processing" on page 8-18.

- Output:
  - Internal MS protocol boundary A (Send NMVT)

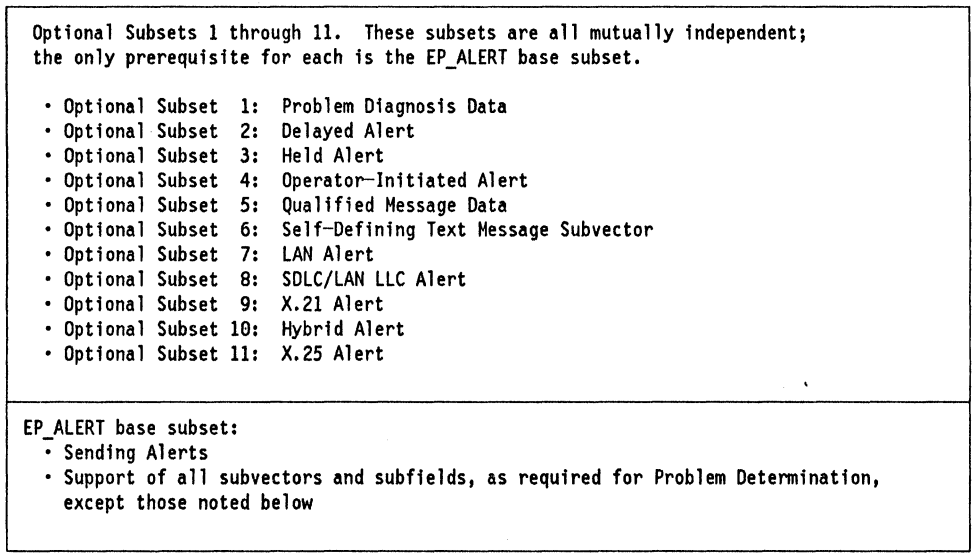
    The EP_ALERT function set group requests the SEND_DATA_SSCP_PU function set group to send an NMVT RU on the SSCP-PU session with its controlling CP. The output data consists of a complete NMVT built by this function set group. The details of this protocol boundary are described in "Protocol Boundary A - Send NMVT" on page 8-18.
  - Internal MS protocol boundary C (Delayed Alert Processing)

    If EP_ALERT optional subset 2 (Delayed Alert) is implemented, the EP_ALERT function set group receives requests from the SEND_DATA_SSCP_PU function set group to process an NMVT to determine whether it can be transmitted at this time, classified as a delayed Alert to be transmitted at a later time, or logged if it cannot be transmitted at all. After receiving such a request, this function set group will produce output for the SEND_DATA_SSCP_PU function set group indicating whether the NMVT can be transmitted. The details of this protocol boundary are described in "Protocol Boundary C - Delayed Alert Processing" on page 8-18.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets



Note: Support for the following subvectors/subfields is not part of the EP_ALERT base subset; it falls into the indicated optional subsets:

Qualified Message Data (X'01') subfield	Optional Subset 5
Self-Defining Text Message (X'31') subvector	Optional Subset 6
LAN LCS Data (X'51') subvector	Optional Subset 7
LCS Configuration Data (X'52') subvector	Optional Subset 8, 9 or 11
SDLC Link Station Data (X'8C') subvector	Optional Subset 8
Text Message (X'00') subvector	Optional Subset 10
Hierarchy Name List (X'03') subvector	Optional Subset 10
Basic Alert (X'91') subvector	Optional Subset 10
Detail Qualifier (EBCDIC) (X'A0') subvector	Optional Subset 10
Detail Qualifier (hexadecimal) (X'A1') subvector	Optional Subset 10

Figure 10-2. Base and Optional Subsets of EP_ALERT Function Set

EP_ALERT Base Subset

Functions Provided

An Alert is an unsolicited notification of an actual or impending loss of availability of a system resource to an end user. This base subset of EP_ALERT provides the following:

- Detects an Alert condition for any resource controlled by this node
- Constructs an NMVT containing an Alert (X'0000') major vector
- Passes the NMVT to the SEND_DATA_SSCP_PU function set group, to be sent unsolicited, to all control points that control the resource to which the Alert applies

Every implementation of PUMS is required to send Alerts.[1] Support of certain *types* of Alerts, and of certain types of data on Alerts, is optional. Also optional is the saving of Alerts when the SSCP-PU session is unavailable and sending them when the session with the SSCP is re-established. Refer to the optional subsets that follow for a detailed description of the Alert options available to implementations.

Alert data falls into two broad categories: data indicating the nature of the problem being reported, and data identifying the component that has the problem. Some portions of this data are textual; these are displayed to the network operator just as they are received. Others are code points that index particular strings of stored text; these strings of text are displayed in a predetermined manner, upon receipt of the code points that index them.

Refer to "Problem Determination and Problem Diagnosis via Alerting" on page 3-11 for additional details on the function provided by this base subset.

Formats Supported

The following subvectors are constructed by this function set group and included in the Alert major vector. The list here includes only those subvectors supported in the base subset of EP_ALERT; additional subvectors are discussed below, in connection with the optional subsets of EP_ALERT.

- A Date/Time (X'01') or Relative Time (X'42') subvector, depending on the node's time-stamping capabilities
- An SNA Address List (X'04') subvector if the origin of the origin of the Alert condition is an SNA-addressable entity other than the sending PU.
- A Hierarchy/Resource List (X'05') subvector logically identifying the origin of the Alert condition.

  This subvector is constructed if the origin of the Alert condition is not located in the products implementing the PU, and cannot be represented in the SNA Address List subvector.

  This subvector is also constructed if the origin of the Alert condition is a component of the hardware product implementing the PU, and requires operator attention to resolve the problem, e.g., an integrated disk drive that requires a diskette to be changed.
- A Product Set ID subvector identifying the PU sending the Alert.
- A second Product Set ID (X'10') subvector physically identifying the origin of the Alert condition, if the origin of the Alert condition is a hardware or software product and is not one of the products implementing the PU

---

[1] More precisely, every implementation is required to be *capable of* sending Alerts, if this function has been enabled. Implementations are permitted to support *selective generation* of their Alerts, i.e., support an interface that allows some or all of their Alerts to be turned off by operator command. The granularity with which Alerts can be selected for enabling is left up to each implementation of EP_ALERT.

- A Supporting Data Correlation (X'48') subvector if this function set group has preserved supporting data, e.g., a storage dump, to which the Alert must be correlated
- A Generic Alert Data (X'92') subvector
- A Probable Causes (X'93') subvector

The X'92' and X'93' subvectors are included in every Alert. The next five subvectors (X'94' - X'98') are included when appropriate. Support for these subvectors is required for an implementation of PUMS, even though not all of them are included in every Alert. The architecture does not specify *what* problem determination data an Alert sender is required to capture for the different Alert conditions it can detect. What it does require is the following: If an Alert sender has captured some problem determination data, and if this data is of a type that the Alert subvectors can transport, then the sender must include the data in the Alert it sends to report the problem. The architecture does not allow an implementation of PUMS to fail to send problem determination data that it has captured simply because it does not choose to support the subvector in which that data would flow.

- A User Causes (X'94') subvector
- An Install Causes (X'95') subvector
- A Failure Causes (X'96') subvector
- A Cause Undetermined (X'97') subvector
- A Detailed Data (X'98') subvector

Implementation Requirements

The requirements for implementing the EP_ALERT base subset are described by a model consisting of subsets of the physical resources manager LMS, LU LMS and DLC Manager LMS and a subset of management services.

Subsets of the Physical Resources Manager LMS, LU LMS, and DLC Manager LMS:

- Interact with PUMS via MS protocol boundaries 1, 2, and 4 (respectively) as follows:
  - Detect a loss or impending loss of availability of the resources that they are managing, and report this loss of availability to MS such that an Alert major vector can be created.

A Subset of Management Services:

- Provides the following process:
  - Sending Alerts
    - Constructs a complete NMVT containing an Alert (X'0000') major vector. The NMVT is passed to the SEND_DATA_SSCP_PU function set group via protocol boundary A.

Sending Alerts:

This process is started by one of the LMSs when it has detected an Alert condition. When this process is started, it receives data from the LMS describing the Alert condition. It uses this data to construct a complete NMVT containing the Alert (X'0000') major vector. This process then starts the process described in "Sending NMVTs" on page 9-14 to send the NMVT to CPMS.

Subvectors are constructed as follows:

- Date/Time (X'01') or Relative Time (X'42') subvector: Depending on the node's capabilities, this process constructs one of two time stamps. See "Building the Date/Time (X'01') and Relative Time (X'42') Subvectors" on page 10-148 for details.
- SNA Address List (X'04') subvector: If the required data is provided by the LMS, this process constructs an SNA Address List subvector. The LMS provides this data if the origin of the Alert condition is an SNA-addressable entity other than the PU. See "Building the SNA Address List (X'04') Subvector" on page 10-148 for details.
- Hierarchy/Resource List (X'05') subvector: If the required data is provided by the LMS, this process constructs a Hierarchy/Resource List subvector. The LMS provides this data to logically identify (by name and resource type) the origin of the Alert condition for the following cases:
  - The origin of the Alert condition is not located in the products implementing the PU, and cannot be represented in the SNA Address List subvector.
  - The origin of the Alert condition is a component of the hardware product implementing the PU, and requires operator attention to resolve the problem, e.g., an integrated disk drive that requires a diskette to be changed.
- Product Set ID (X'10') subvector: One instance of this subvector, identifying the PU sending the NMVT, is is always constructed by this process. If the required data is provided by the LMS, this process constructs a second Product Set ID subvector as well, to identify the component for which the Alert was generated. See "Building the Product Set ID (X'10') Subvector" on page 10-149 for a description of how the Product Set ID subvector is constructed.

  Note: The LMS provides data for construction of a Product Set ID subvector to physically identify the origin of the Alert condition, if the origin of the Alert condition is a hardware or software product different from nhe products implementing the PU.

  If data to identify a hardware product is not available to the LMS, the following values are passed:
  - Machine type: EBCDIC 0's [X'F0']
  - Other hardware ID fields: binary 0's

Note: It is only in this instance of the Product Set ID, identifying an origin of an Alert condition different from the Alert sender, that these 0 values are allowed. The instance of the Product Set ID that identifies the Alert sender must insert have non-zero values in the subvector.

- Self-Defining Text Message (X'31') subvector: *If optional subset 6 (Text Message) was implemented*, and if the LMS provides a text message, this process constructs a Self-Defining Text Message subvector containing the message text and code points identifying the entity sending the message, the coded character set in which the message is encoded, and the national language in which the message is written. All of this information is supplied by the LMS.

  Refer to optional subset 6 for additional details.

- Supporting Data Correlation (X'48') subvector: If correlation data is provided by the LMS, this process builds a Supporting Data Correlation subvector to uniquely identify data associated with the problem reported by this Alert.

  Note: The LMS provides correlation data if a dump, trace, or log data has been saved at the Alert sender, and is required to resolve the Alert condition.

- LAN Link Connection Subsystem Data (X'51') subvector: *If optional subset 7 (LAN Alert) was implemented*, this process constructs this subvector, including in it data supplied by the DLC Manager LMS.

  Refer to optional subset 7 for additional details.

- LCS Configuration Data (X'52') subvector: *If optional subset 8 (SDLC/LAN LLC Alert), optional subset 9 (X.21 Alert) or optional subset 11 (X.25 Alert) was implemented*, this process constructs this subvector, including in it data supplied by the DLC Manager LMS.

  Refer to optional subsets 8, 9 and 11 for additional details.

- SDLC Link Station Data (X'8C') subvector: *If optional subset 8 (SDLC/LAN LLC Alert) was implemented*, this process constructs this subvector, including in it data supplied by the DLC Manager LMS.

  Refer to optional subset 8 for additional details.

- Generic Alert Data (X'92') subvector: This process always constructs this subvector, as follows:

  — The initiation indicator flag is set based on data supplied by the LMS.

  — If optional subset 2 (Delayed Alert) is not implemented, the held Alert indicator and delayed Alert indicator flags are always set to 0 by this process; if optional subset 2 is implemented, these two flags are set based on information passed to this process by the LMS. See "EP_ALERT Optional Subset 2 (Delayed Alert)" on page 10-11 and "EP_ALERT Optional Subset 3 (Held Alert)" on page 10-14 for details of the conditions under which these indicators are set to 1.

  — The Alert Description Code is filled in, based on a value supplied by the LMS.

  - The Alert ID Number is filled in; see "Identification of Unique Alerts" on page 10-27 for a description of this number and how it is calculated.

- Probable Causes (X'93') subvector: This process always constructs this subvector, from data supplied by the LMS.
- User Causes (X'94'), Install Causes (X'95'), Failure Causes (X'96'), and Cause Undetermined (X'97') subvectors: This process constructs one or more of these subvectors, from data supplied by the LMS.
- Detailed Data (X'98') subvector: If the LMS supplies data for a Detailed Data subvector, this process constructs an instance of it.
- Text Message (X'00'), Hierarchy Name List (X'03'), Basic Alert (X'91'), Detail Qualifier (EBCDIC (X'A0'), and Detail Qualifier (hexadecimal) (X'A1') subvectors: *If optional subset 10 (Hybrid Alert) was implemented*, this process always constructs the Basic Alert subvector, and may construct some or all of the other subvectors listed.

  Refer to optional subset 10 for additional details.

After it has completed building subvectors, this process places the subvectors in an Alert (X'0000') major vector. See "Building a Management Services Major Vector" on page 10-151 for details of this step.

After it completes the Alert major vector, this process builds an NMVT to transport it. See "Building an NMVT" on page 10-151 for the details of how a process builds an NMVT.

After it has completed building an NMVT, this process uses internal MS protocol boundary A to start the process described in "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set) to send the NMVT to CPMS."Protocol Boundary A - Send NMVT" on page 8-18 shows the items provided when starting the "Sending NMVTs" process. This process places the following values in these items:

Item 1: A pointer to the NMVT that it has built

Item 2: "NONE" (This data is unsolicited.)

After this process has started the "Sending NMVTs" process, it terminates.

EP_ALERT Optional Subset 1 (Problem Diagnosis Data)

Functions Provided

EP_ALERT optional subset 1 (Problem Diagnosis Data) provides for the inclusion of Problem Diagnosis data in Alerts. The Detailed Data (X'82') subfield in an Alert is capable of carrying Problem Determination data (e.g., port number), and Problem Diagnosis data, such as an error or malfunction code, that points to a particular element within a product. This optional subset is characterized by use of an already supported subfield to provide an additional function, rather than by support of an additional subvector or subfield, As was the case with Problem Determination data, the Alert architecture says nothing about what Problem Diagnosis data an implementer of this subset is required to provide.

Formats Supported

This optional subset involves support for no additional formats.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 1 (Problem Diagnosis Data) are described by a model consisting of subsets of one or more of the LMSS that initiate Alerts. These LMSS provide the ability to include Problem Diagnosis data in the Alert data that they pass to PUMS.

EP_ALERT Optional Subset 2 (Delayed Alert)

Functions Provided

EP_ALERT optional subset 2 (Delayed Alert) provides for the following:

- Detecting the loss of the session between the PU and its controlling CP
- Constructing an Alert (with the Delayed Alert and Held Alert flags set) indicating the reason the session was lost, and holding that Alert until the session is re-established
- Logging all NMVTs that would have been sent unsolicited while the session with the PU's controlling CP has been unavailable, or passing them to EP_ALERT optional subset 3 (Held Alert) if it has been implemented
- Detecting that the session between the PU and its controlling CP has been re-established, and sending the delayed Alert that has been held while the session has been unavailable

Formats Supported

The EP_ALERT optional subset 1 supports the setting of the delayed Alert flag in the Generic Alert Data (X'92') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 2 (Delayed Alert) are described by a model consisting of a subset of PU configuration services, the physical resources manager LMS and DLC manager LMS, and PU management services.

A Subset of PU Configuration Services:

- Interacts with PUMS via MS protocol boundary 6 as follows:
  - Notifies PUMS (the process described in "Sending Delayed Alerts" on page 10-13) whenever the PU's session with its SSCP has been established

Subsets of the Physical Resources Manager LMS and DLC Manager LMS:

- Interact with PUMS via, respectively, MS protocol boundaries 1 and 3 as follows:
  - Determines that the problem being reported to PUMS has caused the SSCP-PU session to be lost, and indicates this fact to PUMS. When it receives such an indication, the Sending Alerts process within PUMS sets to B'1' both the Delayed Alert and Held Alert flags in the Generic

Alert Data (X'92') subvector. It is entirely the responsibility of the LMS to determine whether an Alert condition has affected the SSCP-PU session. Examples of how an LMS might know this to be the case are (1) an Alert condition that results in the loss of a node's only link, and (2) an Alert condition that results in the loss of all the sessions out of a node. If the LMS cannot determine whether an Alert condition has affected the SSCP-PU session (e.g., the LMS might know that the condition has resulted in the loss of a link, but have no knowledge whether this was the link supporting the SSCP-PU session), then it passes no indication to PUMS, and the Delayed Alert and Held Alert flags are set to B'0'.

A Subset of PU Management Services:

- Provides the following processes:
  - Processing delayed alerts

    Processes an NMVT as follows when requested by the SEND_DATA_SSCP_PU function set group (protocol boundary C):

    - Stores it as a delayed Alert if it qualifies
    - Passes it to EP_ALERT optional subset 3 (Held Alert) if it has been implemented, and if the NMVT cannot be transmitted at this time and cannot be stored as a delayed Alert
    - Passes it back to the "Sending NMVTs" process on "Sending NMVTs" on page 9-14 to be sent on the SSCP-PU session if it can be sent at this time.
- Sending delayed Alerts

  Does the following when requested by PU configuration services (via protocol boundary 21):

  - Removes a delayed Alert from the delayed Alert control block and passes it to the SEND_DATA_SSCP_PU function set group to be sent on the SSCP-PU session

Processing Delayed Alerts:

(See Figure 10-5 on page 10-17 and the accompanying text for a complete description of delayed and held Alert processing.

This process is started by the process described in "Sending NMVTs" on page 9-14 and is passed a completed NMVT. See "Protocol Boundary C - Delayed Alert Processing" on page 8-18 for the details of protocol boundary C (Delayed Alert processing).

This process examines the delayed Alert control block (see Figure 10-3 on page 10-13) to determine whether it can send the NMVT. Either this control block is empty, or it contains a pointer to a single NMVT that carries a delayed Alert.

- If there is a pointer in the delayed Alert control block (the control block is not empty), this process checks for the existence of the held Alert control block.
  - If the held alert control block is present (if optional subset 3 of the EP_ALERT function set is implemented), this process starts the process described in "Sending Held Alerts" on page 10-15, passing a pointer to the completed NMVT in item 1. (See "Protocol Boundary B - Held Alert Processing" on page 8-18.)
  - If the held Alert control block does not exist, this process starts an undefined process to log the NMVT.

  This process then returns to the process "Sending NMVTs" on page 9-14 with a non-zero value in returned item 1 indicating that the NMVT cannot be sent at this time on the SSCP-PU session. (See "Protocol Boundary C - Delayed Alert Processing" on page 8-18.)
- If the delayed Alert control block is empty, this process examines the current NMVT to determine whether it should be sent or placed in the delayed Alert control block. If this NMVT contains an Alert (X'0000') major vector, and if the delayed Alert flag in the Generic Alert Data (X'92') subvector is set to B'1', then the NMVT is stored until the SSCP-PU session on which it is to be sent becomes active. This process places a pointer to the NMVT containing a delayed Alert in the delayed Alert control block, shown in Figure 10-3.
  - If a pointer to the NMVT was placed in the delayed alert control block, this process returns to the process "Sending NMVTs" on page 9-14 with a non-zero value in returned item 1 indicating that the NMVT cannot be sent at this time on the SSCP-PU session. (See "Protocol Boundary C - Delayed Alert Processing" on page 8-18.)
  - If a pointer to the NMVT was not placed in the delayed alert control block, this process returns to the process "Sending NMVTs" on page 9-14 with a zero value in returned item 1 and expects SEND_DATA_SSCP_PU to send the NMVT on the SSCP-PU session. (See "Protocol Boundary C - Delayed Alert Processing" on page 8-18.)

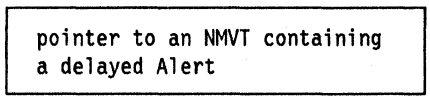


Figure 10-3. Delayed Alert Control Block

Sending Delayed Alerts:

This process sends a delayed Alert, if one is contained in the Delayed Alert Control Block, to the PU's SSCP when that session becomes active.

Whenever an SSCP-PU session to a node becomes active, the PU session manager in the node notifies the appropriate process in PU configuration ser-

vices. The process in PU configuration services then causes this process to be started.

When started, this process examines the delayed Alert control block. If no pointer to an NMVT is present in this control block, then this process terminates. If a pointer to an NMVT (containing a delayed Alert) is there, then this process removes it, leaving the control block empty. It then starts "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set) to send the NMVT. "Protocol Boundary A - Send NMVT" on page 8-18 shows the items provided when starting the "Sending NMVTs" process. This process places a pointer to the NMVT to be sent, into item 1, starts the "Sending NMVTs" process, and then terminates.

EP_ALERT Optional Subset 3 (Held Alert)

Functions Provided

EP_ALERT optional subset 3 (Held Alert) provides the capability to hold Alerts that occur while the session to the control point is out, and to send the Alerts, with an indication that they have been held, after the session is re-established.

Formats Supported

This optional subset supports the setting of the held Alert bit to B'1' in the Flag byte of the Generic Alert Data (X'92') subvector.

Electives

The only elective available to this subset is the choice of the number of Alerts to be held. This number can be chosen to be any value acceptable to the implementation. The implementation holds all Alerts that occur until this value is exceeded, at which time new Alerts are logged, and not held.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 3 (Held Alert) are described by a model consisting of a subset of PU management services.

A Subset of PU Management Services:

- Sending held Alerts
  - Removes a held Alert from the queue (held Alert control block) and passes it to the SEND_DATA_SSCP_PU function set group (via internal MS protocol boundary K) to be sent on the SSCP-PU session (It does this after being started by the SEND_DATA_SSCP_PU function set group via PUMS protocol boundary B.)
  - Places an Alert on the held Alert queue (held Alert control block) when started via MS protocol boundary L by the process described in "Processing Delayed Alerts" on page 10-12, or the process described in "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set). Prior to placing an Alert on the held Alert queue, this process sets the Held Alert and Delayed Alert flags correctly.

Sending Held Alerts:

This process is started to either remove an Alert from the held Alert queue, or to place an Alert on the held Alert queue or log it if queueing is not possible. See "Protocol Boundary B - Held Alert Processing" on page 8-18 for the details of protocol boundary B (Held Alert processing).

This process examines the value passed in item 1 to determine what it is being requested to do. If item 1 is zero, it goes to the *Dequeuing Entry Point*. If item 1 is non-zero, it goes to the *Queuing Entry Point*. Figure 10-4 shows the held Alert control block.

- HEAD_POINTER is a pointer to the first Alert on the queue
- FOOT_POINTER is a pointer to the last Alert on the queue
- COUNT is the number of Alerts in the queue. The maximum number allowed is determined by the elective selected by the implementation.

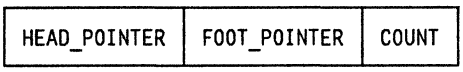


Figure 10-4. Held Alert Control Block

This model chose to chain the Alerts on the queue by overlaying the NS header with a pointer to the next NMVT on the queue. This will be referred to as CHAIN_POINTER. If a three-byte pointer is not adequate for the implementation, a different queueing scheme would be required.

Queuing Entry Point

If the NMVT pointed to by item 1 contains an Alert (X'0000') major vector, this process updates the Delayed and Held Alert flags as follows, to reflect the fact that the Alert has been held. (See Figure 10-5 on page 10-17 for a pictorial representation.)

- If both flags are set to B'1', and if the initiating process was Sending Delayed Alerts (EP_ALERT optional subset 2), then this process resets the Delayed Alert Flag to B'0'. An Alert marked as delayed is passed from Sending Delayed Alerts to Sending Held Alerts only if there is already a delayed Alert in the delayed Alert control block; in this case the delayed flag is no longer appropriate for the current Alert since it was the condition reported by the Alert in the delayed Alert control block, not the condition reported by the current Alert, that prevented the current Alert from being sent to CPMS.
- If both flags are set to B'1', and if the initiating process was Sending NMVTs (SEND_DATA_SSCP_PU base subset), then this process leaves the flags unchanged. An Alert marked as delayed is passed from Sending NMVTs to Sending Held Alerts only if Sending NMVTs was unsuccessful in sending a delayed Alert passed to it by Sending Delayed Alerts. In this case the delayed flag is still appropriate, since the Alert provides information on an

Alert condition that (1) caused the SSCP-PU session to be lost, and (2) was, at least temporarily, recovered from.

- If both flags are set to B'0', then regardless of the initiating process, this process changes the held Alert flag to B'1'. This change indicates that the Alert was held, i.e., was not sent to CPMS immediately upon its creation.
- If the delayed Alert flag is set to B'0' and the held Alert flag is set to B'1', then this process leaves them unchanged. This Alert is one that was previously held, passed to Sending NMVTs for transmission to CPMS, then passed back to Sending Held because it could not be sent. When it finally is sent to CPMS, it should still be marked as a held Alert.

If the NMVT pointed to by item 1 does not contain an Alert (X'0000') major vector, or if the COUNT in the held Alert control block equals the maximum value allowed by the implementation, this process starts an undefined process to log the NMVT and then terminates.

If the NMVT pointed to by item 1 is to be queued, its chain pointer is set to zero. This process then checks the value of FOOT_POINTER. If FOOT_POINTER is zero, the value from item 1 is placed into HEAD_POINTER and FOOT_POINTER. If FOOT_POINTER is non-zero, this process places the value from item 1 into CHAIN_POINTER of the NMVT pointed to by FOOT_POINTER, and overlays FOOT_POINTER with the value from item 1. This process then overlays COUNT with COUNT+1, and terminates.

Dequeuing Entry Point

This process examines CHAIN_POINTER of the NMVT pointed to by HEAD_POINTER. If it points to another NMVT, HEAD_POINTER is overlaid with CHAIN_POINTER, if not, HEAD_POINTER and FOOT_POINTER are set to zero. This process then sets COUNT to COUNT-1, overlays the first three bytes of the NMVT with X'41038D', and starts the process described in "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set) to send the NMVT. "Protocol Boundary A - Send NMVT" on page 8-18 shows the parameters specified when starting the "Sending NMVTs" on page 9-14 process. This process places a pointer to the NMVT to be sent, into item 1. After this process has started the process "Sending NMVTs" on page 9-14, it terminates.

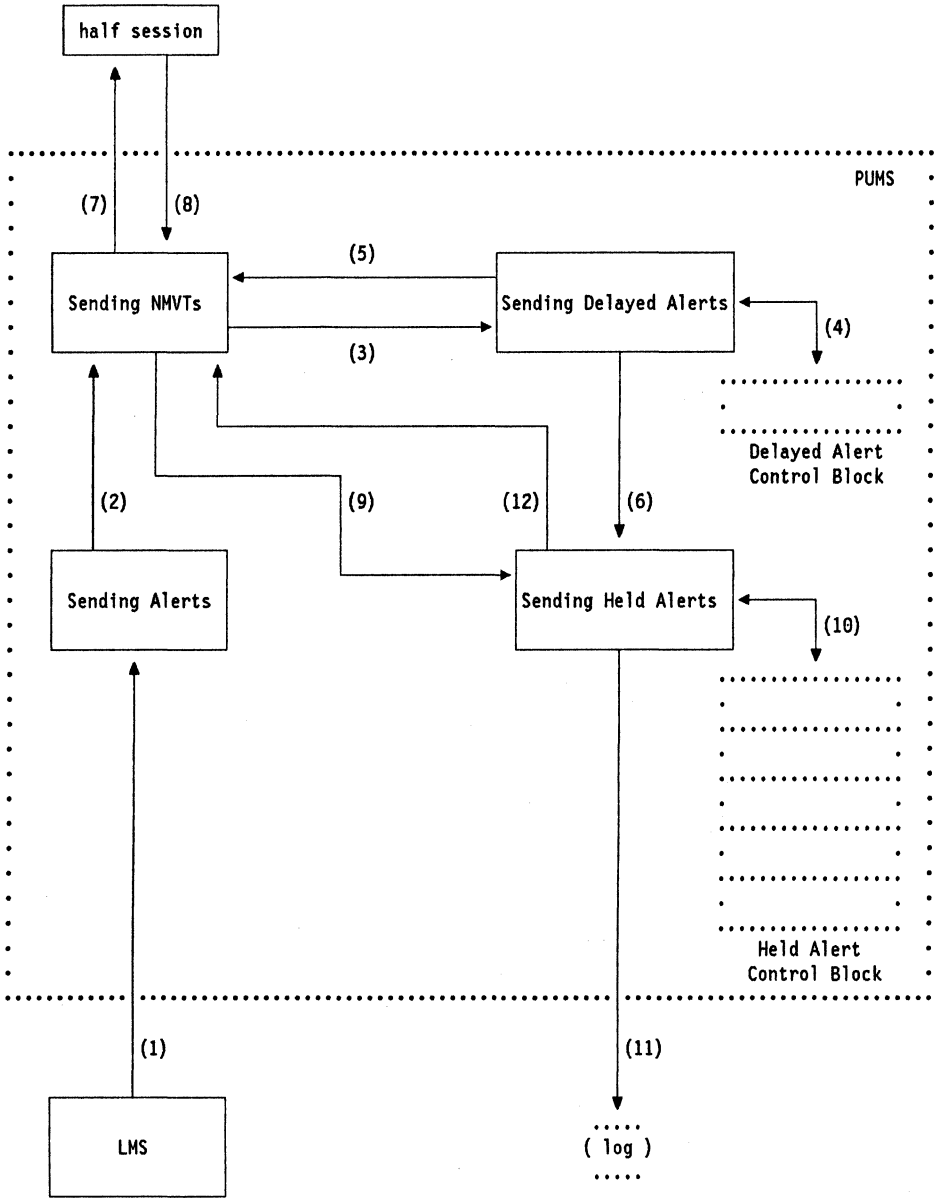


Figure 10-5. Overview of Delayed and Held Alert Processing

Figure 10-5 summarizes delayed and held Alert processing when both the Delayed Alert and Held Alert optional subsets are implemented. The arrows marked (1)-(12) in the figure indicate the following:

- *Arrow 1*: The LMS indicates to the Sending Alerts process (EP_ALERT base subset) that it has detected an Alert condition.
- *Arrow 2*: Sending Alerts uses protocol boundary A (documented in "Protocol Boundary A - Send NMVT" on page 8-18) to pass an Alert major vector encapsulated in an NMVT to Sending NMVTs (SEND_DATA_SSCP_PU base subset).

- *Arrow 3*: Sending NMVTs passes a pointer to this NMVT to Sending Delayed Alerts (EP_ALERT optional subset 2) via protocol boundary C (documented in "Protocol Boundary C - Delayed Alert Processing" on page 8-18).
- *Arrow 4*: Sending Delayed Alerts looks to see if there is already a delayed Alert in the delayed Alert control block.
- *Arrow 5*: If the delayed Alert control block is empty, and if the current Alert is not marked as delayed, then Sending Alerts passes back to Sending NMVTs, via protocol boundary M, an indication that the current Alert may be sent.
- *Arrow 6*: If, in checking the contents of the delayed Alert control block at the request of Sending NMVTs, Sending Delayed Alerts finds that there is already a delayed Alert waiting to be sent, it passes to Sending Held Alerts (EP_ALERT optional subset 3) the pointer that was passed to it by Sending NMVTs. It does this via protocol boundary B, documented in "Protocol Boundary B - Held Alert Processing" on page 8-18.
- *Arrow 7*: After it receives an indication from Sending Delayed Alerts that it should attempt to send an Alert, Sending NMVTs passes the Alert to the half session for transmission to CPMS.
- *Arrow 8*: If the half session is unable to send the Alert, it indicates this failure to Sending NMVTs.
- *Arrow 9*: When it receives such an indication, Sending NMVTs passes to Sending Held Alerts a pointer to the Alert that could not be sent., via protocol boundary B.
- *Arrow 10*: If an Alert was passed to it to be held, Sending Held Alerts looks to see if there is room for it in the held Alert control block. If there is room, it queues the Alert there.
- *Arrow 11*: If the held Alert control block is full, Sending Held Alerts passed the Alert to an undefined process for local logging.

The preceding discussion covers the creation and sending of Alerts, as well as the saving of them as delayed and held. The processes shown in Figure 10-5 also cooperate to send delayed and held Alerts when communication on the SSCP-PU session is restored:

- *Arrows 4 and 5*: After the SSCP-PU session has been restored, PU Configuration Services apprises Sending Delayed Alerts of this fact. Sending Delayed Alerts checks to see if there is currently an Alert in the delayed Alert control block. If there is, it passes a pointer to this Alert to Sending NMVTs, via protocol boundary A.
- *Arrow 7*: Sending NMVTs passes this Alert to the half session.
- *Arrows 8 and 9*: If the half session is again unable to send the Alert, it indicates this fact to Sending NMVTs and, as described above, Sending NMVTs passes a pointer to the Alert to Sending Held Alerts, so that it may be stored in the held Alert control block.
- *Arrows 8 and 9*: If the delayed Alert is successfully transmitted to CPMS, then Sending NMVTs receives a response (positive or negative) from the half

session. At this point it determines whether there is a held Alert control block (i.e., whether EP_ALERT optional subset 3 has been implemented) and, if so, whether this control block currently contains any held Alerts. If there are Alerts currently being held, then Sending NMVTs passes to Sending Held Alerts, via protocol boundary B, an indication that it should remove an Alert from the held Alert control block.

- *Arrows 10 and 12*: Upon receipt of this indication, Sending Held Alerts removes an Alert from the held Alert control block and passes a pointer to it to Sending NMVTs, via protocol boundary A.
- *Arrows 7, 8, and 9*: Sending NMVTs attempts to send this Alerts. If it fails, the Alert is returned to Sending Held Alerts for insertion once again into the held Alert control block.

Note that Sending NMVTs checks for the presence of held Alerts whenever it receives *any* response from CPMS. Thus even if the Held Alert optional subset were implemented without the Delayed Alert optional subset, the process of clearing the held Alert control block would be initiated as soon as any management services data was successfully transmitted to CPMS.

Table 10-1. Setting of the Held and Delayed Alert Indicators by Different Alert Senders

Alert sender:	**H=0,D=0**	**H=0,D=1**	**H=1,D=0**	**H=1,D=1**
Supports neither held Alerts nor delayed Alerts	Always sent	Never sent	Never sent	Never sent
Supports held Alerts, does not support delayed Alerts	On Alerts sent immediately	Never sent	On all held Alerts	Never sent
Supports delayed Alerts, does not support held Alerts	On Alerts sent immediately	Never sent	Never sent	On all delayed Alerts
Supports both held Alerts and delayed Alerts	On Alerts sent immediately	Never sent	On all held Alerts except delayed Alerts	On all delayed Alerts

Table 10-1 summarizes the setting of the Delayed Alert and Held Alert flags by Alert senders implementing both, one of, and neither of the Delayed Alert and Held Alert optional subsets. Two entries in this table require special comment:

- An Alert sender supporting held Alerts but not supporting delayed Alerts sets H=1,D=0 on *all* held Alerts that it sends. Some of these Alerts may very well report conditions that have caused a loss of communications with CPMS, and so in fact qualify as delayed Alerts, but this Alert sender does not distinguish them from other held Alerts.
- An Alert sender supporting delayed Alerts but not supporting held Alerts sets H=1,D=1 on all delayed Alerts that it sends. Thus even though this sender does not support the sending of held Alerts in general, it nevertheless sets the Held Alert *Indicator* on its delayed Alerts.

EP_ALERT Optional Subset 4 (Operator-Initiated Alert)

Functions Provided

EP_ALERT optional subset 4 (Operator-Initiated Alert) provides a mechanism to allow a human operator to initiate the reporting of an Alert condition. This is used to report problems that are not detected automatically by the system. Examples of the types of problems that may only be detected by humans are: poor print or display quality; a machine making an unusual noise; smoke or the smell of something burning coming from a machine. Often the interface provided to the operator will include an area for the operator to enter text describing the Alert condition; if this is the case, the text will flow in the Self-Defining Text Message (X'31') subvector, and optional subset 6 will also be supported. It is possible, however, to support operator-initiated Alerts without supporting user-entered text on these Alerts, by, perhaps, providing the operator with a menu for choosing among the various code points defined by the architecture to describe the Alert condition. This is why support of the Self-Defining Text Message subvector falls into a different optional subset from that into which support of operator-initiated Alerts falls.

Formats Supported

The EP_ALERT optional subset 4 supports the setting of the initiation indicator bit to B'1' in the Flag byte of the Generic Alert Data (X'92') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 4 (Operator-Initiated Alert) are described by a model consisting of a subset of the LU LMS. The LMS provides a mechanism to allow a human operator to report Alert conditions to PUMS. These Alerts are reported to PUMS by providing an indication that the Alert was initiated by an operator. This indication is provided at internal MS protocol boundary 12 along with the data required by the base subset.

The LMS is responsible for ensuring that all required data is provided, just as though the LU itself had detected the problem.

EP_ALERT Optional Subset 5 (Qualified Message Data)

Functions Provided

EP_ALERT optional subset 5 (Qualified Message Data) provides for an alternative method of indexing text messages at an Alert receiver. It is especially tailored for the case in which the Alert sender and the Alert receiver are implemented by different instances of the same product, since in such a case (1) the Alert sender can send to the Alert receiver, via the Qualified Message Data subfield, the same index, and the same qualifier data, that it uses to construct the message that notifies its local operator of the existence of the Alert condition, and (2) the Alert receiver will already have the required messages available, since they are needed for notifying its own operator of Alert conditions that it detects locally.

Formats Supported

This optional subset supports the use of the Qualified Message Data (X'01') subfield within the Detailed Data (X'98') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 5 (Qualified Message Data) are described by a model consisting of a subset of the physical resources manager LMS, LU LMS, and DLC LMS. These LMSs provide the ability to specify an index identifying a particular message stored at the Alert receiver, as well as one or more variable length text strings to be inserted into the message text by the Alert receiver.

See "The Qualified Message Data (X'01') Subfield" on page 10-40 for details.

EP_ALERT Optional Subset 6 (Text Message)

Functions Provided

EP_ALERT optional subset 6 (Text Message) provides the capability to send in an Alert a language-dependent string of text of up to 236 characters. In addition to the text string itself, the Alert identifies the coded character set in which the string is encoded, its national language, and the origin of the text string (e.g., operator, application program).

Formats Supported

This optional subset supports the use of the Self-Defining Text Message (X'31') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 6 (Text Message) are described by a model consisting of a subset of the physical resources manager LMS, LU LMS, and DLC LMS. These LMSs provide the ability to add to an Alert a text message and indications of its coded character set, its national language, and the nature of its origin. See "Text in Alerts" on page 10-26 for details.

EP_ALERT Optional Subset 7 (LAN Alert)

Functions Provided

EP_ALERT optional subset 7 (LAN Alert) provides the capability to send Alerts for errors detected at the MAC layer of a token-ring, CSMA/CD, or bridged LAN. See "Alerts for Local Area Networks" on page A-1 for a list of the Alerts defined for this optional subset.

Formats Supported

This optional subset supports the use of the LAN Link Connection Subsystem Data (X'51') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 7 (LAN Alert) are described by a model consisting of a subset of the DLC manager LMS. This LMS provides the ability to send the appropriate Alerts from "Alerts for Local Area Networks" on page A-1, depending on the type and LAN role of the node in which the DLC manager resides.

EP_ALERT Optional Subset 8 (SDLC/LAN LLC Alert)

Functions Provided

EP_ALERT optional subset 8 (SDLC/LAN LLC Alert) provides the capability to send Alerts for problems detected on SDLC and LAN LLC logical connections. See "SDLC/LAN LLC Alerts" on page A-40 for a list of the Alerts defined for this optional subset.

Formats Supported

This optional subset supports the use of the Link Connection Subsystem Configuration Data (X'52') and SDLC Link Station Data (X'8C') subvectors.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 8 (SDLC/LAN LLC Alert) are described by a model consisting of a subset of the DLC manager LMS. This LMS provides the ability to send all of the Alerts from one or both of the sections "LAN LLC Alerts" on page A-41 and "SDLC Alerts" on page A-55, depending on the nature of the link connections in question.

EP_ALERT Optional Subset 9 (X.21 Alert)

Functions Provided

EP_ALERT optional subset 9 (X.21 Alert) provides the capability to send Alerts for problems detected on X.21 (including X.21 Short Hold Mode) link connections. See "X.21 and X.21 Short Hold Mode Alerts" on page A-69 for a list of the Alerts defined for this optional subset.

Formats Supported

This optional subset supports the use of the Link Connection Subsystem Configuration Data (X'52') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 9 (X.21 Alert) are described by a model consisting of a subset of the DLC manager LMS. This LMS provides the ability to send all of the Alerts from "X.21 and X.21 Short Hold Mode Alerts" on page A-69.

EP_ALERT Optional Subset 10 (Hybrid Alert)

Functions Provided

EP_ALERT optional subset 10 (Hybrid Alert) provides the capability to send a single Alert record that can be processed by both new and old implementations of CPMS. ("New" versus "old" Alert receivers are distinguished by the subvectors they require in an Alert: an old Alert receiver requires the Basic Alert (X'91') subvector, while a new Alert receiver requires the Generic Alert Data (X'92') subvector.) Since the parsing rules for Management Services major vectors specify that all unrecognized subvectors are ignored, the older implementations will process only the older subvectors listed below. Thus a hybrid Alert is, for these implementations, indistinguishable from an Alert containing only the older subvectors.

A new implementation, on the other hand, processes the new subvectors, described in the base subset and the other optional subsets. Either it does not recognize these older subvectors at all (and thereby ignores them), or it is coded to "prefer" the new subvectors to the old ones. To simplify this second option, a single constraint is introduced on the relative positioning of the old and new subvectors; see "Implementation Requirements" below.

Formats Supported

This optional subset supports the use of the following subvectors:

- Text Message (X'00')
- Hierarchy Name List (X'03')
- Basic Alert (X'91')
- Detail Qualifier (EBCDIC) (X'A0')
- Detail Qualifier (Hexadecimal) (X'A1')

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 10 (Hybrid Alert) are described by a model consisting of a subset of the physical resources manager LMS, LU LMS, DLC LMS, and PU Management Services. These components create the referenced subvectors in precisely the way documented in earlier editions of *SNA Format and Protocol Reference Manual: Management Services* (SC30-3346-0 and SC30-3346-1). The subvectors are then included in the Alert major vector with the subvectors created by the base subset and the other optional subsets. If the Hierarchy Name List (X'03') subvector is included in the Alert major vector, it must appear *after* the Hierarchy/Resource List (X'05') subvector. Otherwise there are no constraints on the relative ordering of the subvectors created by this subset and those created by the base subset and the other optional subsets.

EP_ALERT Optional Subset 11 (X.25 Alert)

Functions Provided

EP_ALERT optional subset 11 (X.25 Alert) provides the capability to send Alerts for problems detected on X.25 link connections. See "Alerts for X.25 Link Connections" on page A-88 for a list of the Alerts defined for this optional subset.

Formats Supported

This optional subset supports the use of the Link Connection Subsystem Configuration Data (X'52') subvector.

Implementation Requirements

The requirements for implementing the EP_ALERT optional subset 11 (X.25 Alert) are described by a model consisting of a subset of the DLC manager LMS. This LMS provides the ability to send all of the Alerts from "Alerts for X.25 Link Connections" on page A-88.

Details of the Alert Encodings

This section contains detailed discussions of several topics related to the encoding of the Alert subvectors. You should refer to the Alert subvectors in *Systems Network Architecture Formats*, GA27-3136, as you read this section.

Default/Replacement Code Points

There are six types of code points in a generic Alert that share the same structure and semantics:

- Alert Description Code, in the X'92' subvector
- Probable Causes, in the X'93' subvector
- User Causes, in the X'94' subvector
- Install Causes, in the X'95' subvector
- Failure Causes, in the X'96' subvector
- Recommended Actions, in the X'81' subfield

These code points are termed *default/replacement* code points; their structure is represented in Figure 10-6.

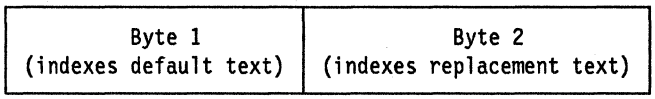


Figure 10-6. Format for Default/Replacement Alert Code Points

Each code point is two bytes long, with the second byte serving to qualify the information carried in the first byte. A code point with a value of X'00' in the second byte is termed a *default* code point. It corresponds to a high-level description of an Alert condition, cause or recommended action. A default code

point is typically followed by one or more *replacement* code points, that have the same first byte but a non-zero second byte. These code points correspond to more specific descriptions, that fall under the high-level description indexed by the default code point. Default/replacement code points are illustrated in Table 10-2. The specific descriptions "RUN CONSOLE TEST," etc. can be seen to fall under the more general description "RUN APPROPRIATE TEST."

Table 10-2. Default and Replacement Text for Recommended Action Code Points

Code Point	Default Text	Replacement Text
X'0400'	RUN APPROPRIATE TEST	RUN APPROPRIATE TEST
X'0401'	" "	RUN CONSOLE TEST
X'0402'	" "	RUN CONSOLE LINK TEST
X'0403'	" "	RUN MODEM TESTS

Default/replacement code points allow for certain flexibility in displays at an Alert receiver. When it receives a replacement code point, an Alert receiver is allowed to display either the text for this code point itself or the text for the default code point above it. Upon receipt of a Recommended Action of X'0401', for example, an Alert receiver may display either "RUN CONSOLE TEST" or the more general "RUN APPROPRIATE TEST." There are two reasons why an Alert receiver might choose to display default text when it receives a replacement code point:

- If a new replacement code point has been added to the architecture and is being sent by an Alert sender, but the receiving product has not yet had its tables updated, then it can still communicate some information by displaying the default text. Suppose, for example, that a new Recommended Action code point X'0404', "RUN TEXT-X TEST," has been added to the architecture. If an Alert receiver receives this code point before it has updated its tables to include the new text, it can still display the default text "RUN APPROPRIATE TEST." Later, after its table is updated, it will be able to display the more informative "RUN TEST-X TEST," but in the interim it has been able to extract useful information from the code point and display this information to a network operator without having had to synchronize its table update with the release of the new sending product.
- An Alert receiver running on a small system might elect to display *only* default text, in order to keep the size of its tables small. Or it might elect to display only default text for some code points, perhaps User, Install, and Failure Causes, while maintaining the full tables of default and replacement text for Alert Description, Probable Causes and Recommended Actions. In any case, default/replacement code points allow an Alert sender to send identical Alerts to Alert receivers of different capabilities.

An additional advantage of default/replacement code points follows simply from the fact that they are code points: An Alert receiver can provide national language support for them, by providing different tables of text for the different languages. It is even possible for a receiver to provide full tables of default and replacement text in certain languages, but to provide shorter tables of default text only for others.

Default/replacement code points for User, Install, and Failure Causes, and for Recommended Actions, carry an additional piece of information in the way they are encoded. Since these code points provide for the insertion of strings of qualifier text, flowing in the Detailed Data (X'82') subfield, or for the insertion of product data, identified in the Product Set ID Index (X'83') subfield, into the stored text that the code points index, the code points must themselves specify implicitly how many X'82'/X'83' subfields are to be associated with them by the receiver processing the Alert. See "Associating Code Points with X'82' and X'83' Subfields" on page 10-37 for further details.

Text in Alerts

Textual data is carried in Alerts in three places:

- In the Detailed Data (X'82') common subfield
- In the Self-Defining Text Message (X'31') common subvector
- In other management services common subvectors:
  - Hierarchy/Resource List (X'05')
  - Product Set ID (X'10')

Textual data from all of these sources may be combined by an Alert receiver to create a single display.

To specify fully how a field containing textual data is to be interpreted, three pieces of information about the field are needed:

- The *character set* used in it.
- The *code page* according to which the data in it is encoded.

These first two pieces of information jointly define the *Coded Graphic Character Set* for the field in question.

- The *symbol strings* allowed in it.

Thus the following data might be specified for a particular field:

- It can contain any upper-case letters and/or numerals. (This by itself says nothing about the bit patterns that represent the characters.)
- It is encoded according to code page xxx of some standard. (This correlates characters with bit patterns, but the only limit it places on the character set is the full set of characters represented on code page xxx.)

Taken together, the first two pieces of information say that the field contains upper-case letters and numerals, encoded as they are on code page xxx. This defines the Coded Graphic Character Set for the field.

- The first character in any instance of the field must be either a numeral or an 'X'. (This characterizes the strings that can appear in the field, rather than the individual characters.)

In the Alert architecture, the Detailed Data (X'82') subfield contains a 1-byte Data Encoding field indicating how an Alert receiver is to display the detailed

data. The following values of the Data Encoding field are defined for the Detailed Data subfield:

X'00': Hexadecimal. The detailed data is treated as a hexadecimal number. If, for example, the detailed data is X'0107', then "0107" is displayed.

X'01': Binary. The detailed data is treated as an unsigned integer, for which the decimal equivalent is displayed. In this case the detailed data X'0107' results in the display "263."

X'11': Coded Graphic Character Set 00640-00500 plus three additional characters. This Coded Graphic Character Set is documented in the *Systems Network Architecture Formats*, GA27-3136. It contains 26 upper and lower case letters, 10 numerals, and 19 special characters. Data Encoding X'11' refers to the union of this Coded Graphic Character Set with the following three characters and hexadecimal values:

"$" - dollar sign: X'5B'

"#" - number sign: X'7B'

"@" - at sign: X'7C'

Coded Graphic Character Set 00640-00500 is displayable on the vast majority of ASCII and EBCDIC terminals, and capable of being entered from the vast majority of ASCII and EBCDIC keyboards. The three additional characters are included for migration; they are present in the old SNA Character Set A, and thus may appear in such SNA designators as LU or CP name.

The Self-Defining Text Message (X'31') subvector potentially supports *any* Coded Graphic Character Set, so the mechanism for identifying the Coded Graphic Character Set in the X'31' subvector must be a general one. Thus this subvector carries a 4-byte field, with 2 bytes identifying a character set and two bytes identifying a code page.

The Self-Defining Text Message subvector also carries an indication of the national language of the text message being transported. With this information, an Alert receiver is able to route messages to different operators based on the set of languages that each operator understands.

Identification of Unique Alerts

One requirement on the generic Alert architecture is that a mechanism be provided for identifying individual Alerts, so that customers can alter the presentation for selected Alerts. This requirement is met via the *Unique Alert Identifier*, which, as Figure 10-7 on page 10-28 indicates, consists of two parts, the Product ID and the Alert ID Number.

Unique Alert Identifier:

Product ID (from the Alert sender's PSID)	Alert ID Number (from the Generic Alert Data (X'92') SV)

Figure 10-7. The Unique Alert Identifier

The Alert ID Number, which flows as bytes 7-10 of the Generic Alert Data (X'92') subvector, serves to differentiate the Alerts sent by one Alert sender. Also, since an Alert ID Number serves to summarize the contents of the fields in an Alert that describe the Alert condition being reported, an Alert ID Number may be associated with a particular Alert condition without regard to the product sending the Alert. This latter feature of Alert ID Numbers is especially significant in an environment that features "common building blocks," i.e., units of hardware, microcode, or software that are incorporated into different products. An Alert condition detected and reported by such a common building block will result in Alerts with identical Alert ID Numbers being sent by each of the different products incorporating the common building block: the Alert ID Number will make it possible for an operator (or application) at an Alert receiver to identify these Alerts from different products as ones that are in fact reporting different instances of the same Alert condition.

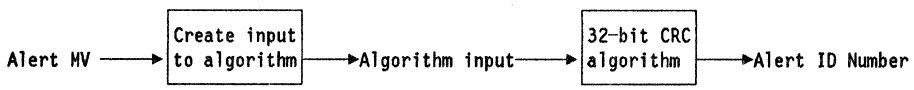


Figure 10-8. Two-Stage Procedure for Generating an Alert ID Number

Figure 10-8 shows the two-stage procedure by which the Alert ID Number is generated for a particular Alert. Briefly, the procedure involves first extracting from the Alert major vector, in a specified order, the contents of the Alert that are significant for defining the Alert ID Number, and then passing these contents through a specified 32 bit cyclic redundancy check (CRC) algorithm to generate a 4 byte Alert ID Number. This number then flows in the Alert major vector from the Alert sender to the Alert receiver.

The input to the CRC algorithm consists of the following code points from the Alert major vector, beginning at the left of the input:

1. The Alert Type
2. The Alert Description Code
3. All Probable Causes code points, in order
4. The delimiter X'FFFF'

This delimiter serves in three places to separate groups of causes code points from each other. Without it there would be no way to tell whether a particular code point was, e.g., the last Probable Cause or the first User Cause.

5. All User Causes code points, in order, if any are present
6. The delimiter X'FFFF'
7. All Install Causes code points, in order, if any are present
8. The delimiter X'FFFF'
9. All Failure Causes code points, in order, if any are present

Note that the Recommended Action code points in an Alert do not figure in the computation of its Alert ID Number. These code points are not included because they do not necessarily relate to the Alert condition per se; they may, for example, vary depending on whether the Alert sending node is attended or unattended, whether it is locally or remotely attached to the Alert receiver, or whether the product implementing it is serviced by the customer or by the vendor.

$$\frac{x^{32}I(x) + x^{k}L(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

where:

L(x) $\sum_{i=0}^{31} x^i$

G(x) $\sum_{i} x^i$ for i = 32,26,23,22,16,12,11,10,8,7,5,4,2,1,0

I(x) The polynomial represented by the input to the CRC algorithm

k number of bits in the input polynomial I(x)

The Alert ID number is the complement of the remainder polynomial R(x) (sometimes represented as *Alert ID* = $\overline{R(x)}$). The reader should remember that all arithmetic is modulo 2, and that the degree of the remainder polynomial, R(x), is less than 32.

The CRC formula on page 10-29 shows the equations defining the algorithm that actually generates the 32 bit Alert ID Number. The input to the algorithm represents the coefficients of the terms of the polynomial I(x), in order of *decreasing* powers of x. For example, the first bit of the Alert Type code represents the coefficient of the x^k term of I(x), the second bit of the Alert Type code represents the coefficient of the x^{k-1} term, and so on. The CRC formula on page 10-29 summarizes the operations performed on the input to the algorithm:

Operation	*Usual Implementation*
Multiply I(x) by x^{32}:	Shift the input 32 places to the left.
Add to this x^k L(x):	Complement the first 32 bits of the input.
Divide this by G(x):	Perform polynomial division.
Complement the remainder polynomial:	(self-explanatory)

The result of these operations is the Alert ID Number.

The 32 bit CRC algorithm documented here is identical to that specified in the ISO International Standards 8802/3 (CSMA/CD) and 8802/5 (Token Ring Access Method) for implementing CRCs on local area networks.

An Alert sending product has two alternatives for determining the Alert ID Numbers to be included the Alerts that it sends:

- The product may implement both of the stages shown in Figure 10-8 on page 10-28 in its code; such senders will generate their Alert ID Numbers dynamically, as each Alert is being prepared for transmission.
- The product's developers may manually catalogue the relevant code points for each of the Alerts that the product will send (i.e., they may perform the first stage from Figure 10-8 manually). These code points then serve as input to an offline implementation of the CRC formula on page 10-29; the resulting Alert ID Numbers are then stored in a table, to be retrieved and inserted in each Alert as it is sent.

When it receives an Alert, an Alert receiver creates the Unique Alert Identifier for it by combining the Alert ID Number with the identity of the Alert sender. Thus an Alert might be identified as "9988 Alert 15F3 2684," or "8899 Alert 3392 483B," where 9988 and 8899 denote fictitious IBM machine types. The single product to be identified as *the* Alert sender for the purposes of creating this unique identifier is always that identified in the first Product Identifier (X'11') subvector within the first (i.e., Alert sender's) Product Set ID.

An Alert receiver can make the Unique Alert Identifier available to a user application program before it creates its own display for the Alert. If the user has written an alternative presentation for the Alert, the user application can pass this display to the Alert receiver together with an indication that the normal display is not to be created for this Alert. Otherwise the user application indicates to the Alert receiver that the usual display is to be created.

Reserved Ranges of Alert Code Points

There are two different ranges of code points that are reserved in the Alert architecture, for different reasons:

- *Code points reserved for non-IBM use:* These are all code points of the form X'Exxx' in the following sets
  - Alert Description
  - Probable Cause
  - User Cause
  - Install Cause
  - Failure Cause
  - Recommended Action

  as well as all Data ID code points of the form X'Ex'. These code points will never be sent by any IBM Alert sender, and they will never be defined in the Management Services Alert architecture. They are reserved for the use of OEMs, VARs, customer-written applications, etc.

Neither Systems Network Architecture nor IBM makes any attempt to regulate the use of code points within this range by different non-IBM implementations.

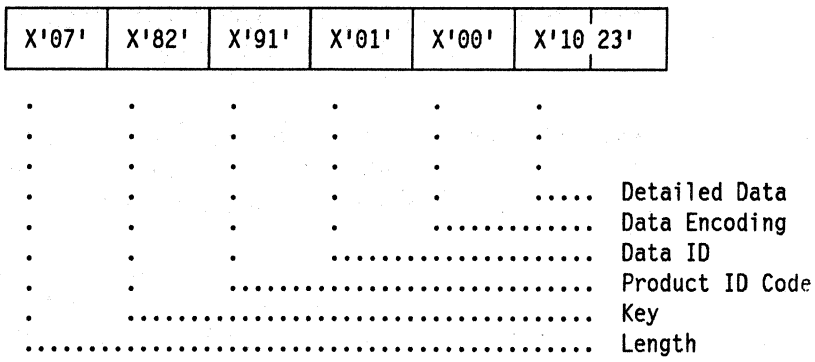
- *Code points reserved for indexing possible future subfield(s):* The User Cause, Install Cause, Failure Cause, and Recommended Action code points currently employ a scheme whereby the third hexadecimal digit of each code point indicates whether subsequent X'82' or X'83' subfields are to be associated with the code point; see "Associating Code Points with X'82' and X'83' Subfields" on page 10-37 for details. Currently this scheme employs third digits from X'A' to X'E'. The remaining possible third digit, X'F', is currently reserved. This has been done so that if a requirement is later identified for an additional subfield to be associated with the "Causes" and Recommended Action code points, an easy migration path will still be available.

The Detailed Data (X'82') Common Subfield

Each instance of the Detailed Data subfield corresponds to a single unit of display (i.e., a single body of information that is displayed as a unit) at the Alert receiver. In specifying this unit of display, the subfield requires four separate pieces of data:

- *Product ID Code:* This field instructs the Alert receiver as to what product data, if any, is to be included in the unit of display. It contains the same encoding structure as the Product Set ID Index (X'83') subfield. See Table 10-3 on page 10-36 for a description of this structure.
- *Data ID:* This field contains a code indicating the type of data present in the subfield. These codes index text, identifying the type of the data, to be included in the display.
- *Data Encoding:* This field contains a code indicating how the data in the subfield is actually encoded, and thus how it is to be displayed. See "Text in Alerts" on page 10-26 for further details.
- *Detailed Data:* This field contains the detailed data itself, encoded as the value in the Data Encoding field specifies.

Detailed Data (X'82') Subfield:



Unit of Display:

Figure 10-9. Example of the Detailed Data (X'82') Subfield and the Corresponding Unit of Display

Figure 10-9 shows an example of the Detailed Data subfield and a possible unit of display corresponding to it. The unit of display is constructed as follows by the Alert receiver:

- "ACF/FICT": The Product ID Code, X'91', instructs the Alert receiver to display the Software Product Common Name for the first software product identified in the Alert sender's Product Set ID. The example assumes that this Alert was sent by FICT, the fictitious software product ACF/FICT.
- "ABEND CODE": The Data ID, X'01', indexes this text at the Alert receiver.
- "1023": The Data Encoding, X'00', instructs the Alert receiver to interpret the detailed data as a hexadecimal number. Thus the detailed data X'1023' is displayed as "1023."

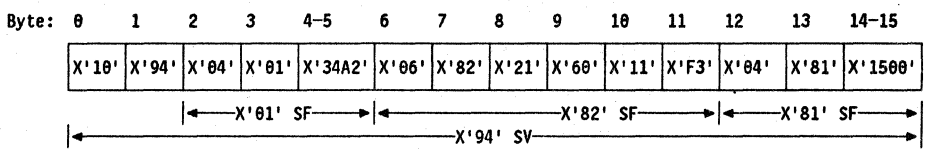
The Detailed Data subfield can occur in five Network Alert subvectors: The Detailed Data (X'98') subvector itself, and the four "Causes" subvectors (X'94'-X'97'). When this subfield occurs in one of the "Causes" subvectors, the unit of display that is constructed from the data it contains appears within the display created from the subvector itself, at a place indicated by the expression "(sf82 qualifier)." This expression may occur at any place within the text associated with a "Causes" or Recommended Action code point: at the beginning, in the middle, or at the end. Up to three instances of this expression may occur within the text associated with any one code point. The following examples illustrate the placement of these points:

X'F0A3' FAILURE OCCURRED ON (sf82 qualifier)
X'20A0' NO RESPONSE FROM THE X.21 NETWORK — (sf82 qualifier) EXPIRED
X'12C0' RETRY AFTER (sf82 qualifier) (sf82 qualifier)
X'32D1' LOCAL DCE COMMUNICATIONS INTERFACE (sf82 qualifier) (sf82 qualifier) (sf82 qualifier)

See "Associating Code Points with X'82' and X'83' Subfields" on page 10-37 for a discussion of how a X'82' subfield is associated with the correct causes or Recommended Action code point in a X'94'-X'97' subvector.

When the subfield occurs in the Detailed Data subvector, the unit of display is either incorporated into a qualified message display or presented independently. See "The Qualified Message Data (X'01') Subfield" on page 10-40 for a discussion of the architecture for qualified messages.

The following example illustrates the use of the Detailed Data subfield within a "Causes" subvector. Figure 10-10 on page 10-34 shows a User Causes (X'94') subvector that might be sent to report an unplugged cable. (In the example the machine type of the Alert sender is 9988.) Figure 10-11 on page 10-34 shows a possible unit of display resulting from this subvector.




```
Interpretation:

Byte  0      (X'10'):  Length of X'94' SV
Byte  1      (X'94'):  SV key
Byte  2      (X'04'):  Length of X'01' SF
Byte  3      (X'01'):  SF key
Byte  4-5  (X'34A2'):  User cause code point X'34A2':
                             CABLE UNPLUGGED:  (sf82 qualifier)
Byte  6      (X'06'):  Length of X'82' SF
Byte  7      (X'82'):  SF key
Byte  8      (X'21'):  PSID index:  Machine type from HW X'11', Alert sender PSID, SV #1
Byte  9      (X'60'):  Code Point:  PORT NUMBER
Byte 10      (X'11'):  Code Point:  Character Set 00640-00500 plus '$', '#', and '@'.
Byte 11      (X'F3'):  EBCDIC '3'
Byte 12      (X'04'):  Length of X'81' SF
Byte 13      (X'81'):  SF key
Byte 14-15 (X'1500'):  Recommended action code point X'1500':
                                CORRECT INSTALLATION PROBLEM
```

Figure 10-10. Example Encoding of the User Causes (X'94') Subvector

```
.
.
.
USER CAUSED - CABLE UNPLUGGED: 9988 PORT NUMBER 3
    ACTIONS - CORRECT INSTALLATION PROBLEM
.
.
.
```

Figure 10-11. Possible Unit of Display for the Specified X'94' Subvector

The individual elements of the display in Figure 10-11 were derived as follows:

- "USER CAUSED-": From the presence of the X'94' subvector
- "CABLE UNPLUGGED:": From user cause code point X'34A2'
- "9988": Hardware machine type of the Alert sender; retrieved from the first Product Set ID (X'10') subvector, because of product ID code point (X'21')
- "PORT NUMBER": From data ID code point X'60'
- "3": EBCDIC data, that flowed in the X'82' subfield
- "ACTIONS-": From the presence of the X'81' subfield
- "CORRECT INSTALLATION PROBLEM": From recommended action code point X'1500'.

The formatting of the unit of display shown in Figure 10-11 is just for the purposes of illustration. Architecturally, the only requirement is that all of the information presented there be displayed as a unit by an Alert receiver.

Product Set ID Indexing

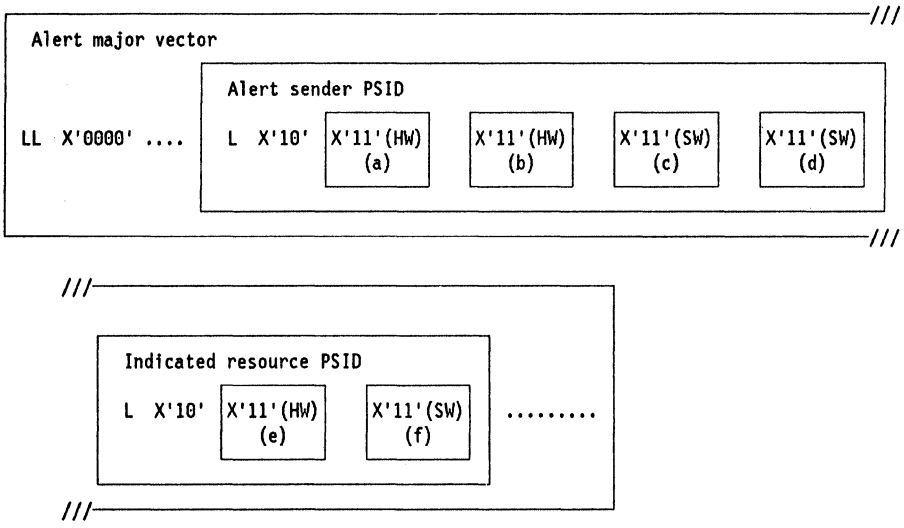
Product identification data flowing in Alerts is displayed in three ways by an Alert receiver:

- Separately, as a part of a Product Set ID display for either the Alert sender or the indicated resource.
- In conjunction with strings of detailed data.
- In conjunction with the text indexed by certain "Causes" or Recommended Action code points.

For the second and third of these uses, the Alert sender must be able to specify a particular Product ID (X'11') subvector, and particular data within this X'11' subvector, to be displayed in conjunction with a given text string. This is accomplished by an encoding structure that appears in three places: in the Product Set ID Index (X'83') subfield, in the Product ID Code field of the X'82' subfield, and in the Product ID Code field of the Qualified Message Data subfield (if EP_ALERT optional subset 5 (Qualified Message Data) is implemented). This structure is shown in Table 10-3 on page 10-36.

Table 10-3. One-byte Structure for Indexing a Product Identifier (X'11') Subvector	
Bits	**Meaning**
Bits 0-3	Product ID subvector code: X'0': no product identification data is to be displayed X'2': hardware X'11', machine type (hardware product common name if present) X'3': hardware X'11', serial number or repair ID number X'4': hardware X'11', machine type (hardware product common name if present) plus serial number or repair ID number X'5': hardware X'11', machine type (hardware product common name if present) plus model number X'6': hardware X'11', machine type (hardware product common name if present) plus model number plus serial number or repair ID number X'9': software X'11', software product common name
Bit 4	Product Set ID indicator: B'0': Alert sender Product Set ID B'1': Indicated resource Product Set ID
Bits 5-7	Count: A three-digit binary number indicating which Product ID subvector is being indexed

The first field here contains a single hexadecimal digit that functions as a code point. It specifies the *type* of X'11' subvector, i.e., hardware or software, to be indexed, and also the data within the subvector to be indexed. The second field is a single bit, indicating whether the X'11' subvector to be indexed is in the Alert sender (first) or the indicated resource (second, or only) Product Set ID. The remaining three bits comprise a binary count, indicating which X'11' subvector, of the type indicated in the first field and in the PSID indicated in the second field, is to be used. Only X'11' subvectors of the type specified in the first field are counted. Figure 10-12 on page 10-37 gives examples showing how this counting is done.



Product ID Subvector Code	PSID	Count	X'11' SV Indexed
X'2', X'3', or X'4' (HW)	B'0'	B'001'	(a)
X'2', X'3', or X'4' (HW)	B'0'	B'010'	(b)
X'9' (SW)	B'0'	B'001'	(c)
X'9' (SW)	B'0'	B'010'	(d)
X'2', X'3', or X'4' (HW)	B'1'	B'001'	(e)
X'9' (SW)	B'1'	B'001'	(f)

Figure 10-12. Examples Illustrating How X'11' Subvectors are Indexed

If the product identification text is to be displayed in conjunction with a "Causes" or Recommended Action code point, it is introduced into the stored text indexed by that code point at a location indicated in the architecture by the expression "(sf83 product text)." The same third digit mechanism that indicates how many Detailed Data (X'82') subfields are associated with a code point also indicates whether a X'83' subfield is associated with it; see "Associating Code Points with X'82' and X'83' Subfields" for details.

Associating Code Points with X'82' and X'83' Subfields

This section describes the mechanism by which the subfields in the X'94'-X'97' subvectors are associated with the appropriate "Causes" and Recommended Action code points. Figure 10-13 on page 10-38 shows a high-level view of one of the "Causes" (X'94'-X'96') subvectors. The subvector in this figure contains the following:

- A "Causes" (X'01') subfield containing three cause code points
- Detailed Data (X'82') subfields (a) - (d), providing substitution text for the text strings indexed by the three cause code points

- A Recommended Actions (X'81') subfield containing four Recommended Action code points
- Detailed Data (X'82') subfields (e) and (f), providing substitution text for the text strings indexed by the four Recommended Action code points

The Cause Undetermined (X'97') subvector differs from the X'94'-X'96' subvectors in that it carries no X'01' subfield. It does, however, carry the X'81' subfield and any X'82' subfields associated with it. Thus there is still a task of associating X'82' subfields with code points in the X'97' subvector, but only for one set of code points rather than for two.

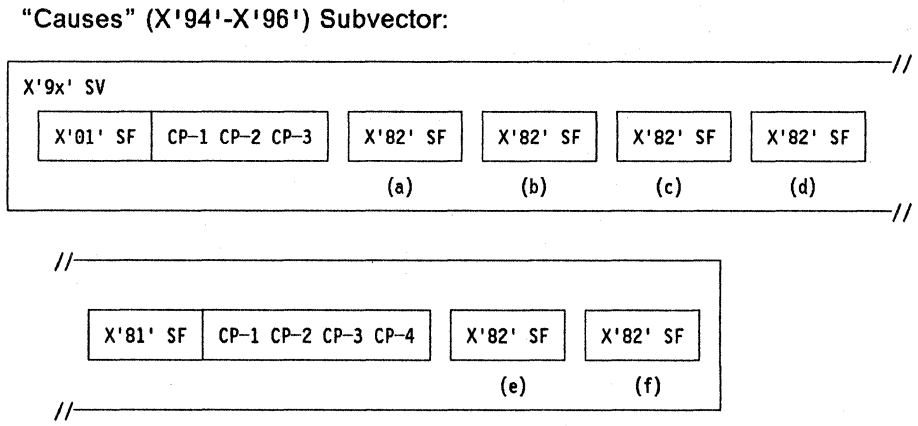


Figure 10-13. Structure of the Subfields within a 'Causes' (X'94'-X'96') Subvector

The association of Detailed Data (X'82') and Product Set ID Index (X'83') subfields with code points is accomplished by means of the code points themselves. The third hexadecimal digit of each code point specifies the number of X'82' subfields associated with that code point, and whether there is a X'83' subfield associated with it, according to the following scheme:

X'xx0x'-X'xx9x':	No X'82' or X'83' subfields.
X'xxAx'-X'xxBx':	One X'82' subfield.
X'xxCx':	Two X'82' subfields.
X'xxDx':	Three X'82' subfields.
X'xxEx':	One X'83' subfield
X'xxFx':	Reserved: code points are not currently assigned in this range.

Note that there is no option for associating multiple X'83' subfields with a code point, or for associating both X'82' and X'83' subfields with a code point.

Given this encoding scheme, the Alert receiver can always make the correct association of code points with X'82' or X'83' subfields, even if the code points are new replacement code points that it does not recognize. In the subvector

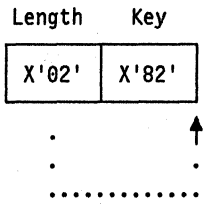
shown in Figure 10-13, for example, if CP-1 in the X'01' subfield is X'2162', the Alert receiver knows that none of the X'82' subfields (a) through (d) is associated with it. If CP-2 in the X'01' subfield is X'25D1', the Alert receiver knows that X'82' subfields (a), (b), and (c) are associated with it. Similar processing is done for all of the code points in the X'01' and X'81' subfields, to correlate all of the X'82' subfields with the appropriate code points.

There are two features of this scheme that require further comment:

- If the Alert receiver receives a replacement code point that it does not have in its table, then it does not have the replacement text that this code point indexes. In this case the architecture specifies that it display the text for the associated default code point. If, however, the unrecognized replacement code point had one or more X'82' subfields associated with it, the text resulting from these subfields will not fit naturally into the text indexed by the default code point. (Since default code points are always of the form X'xx00', and so always have a '0' as their third digit, their text is always designed to have no qualifier text inserted into it.) In this case the Alert receiver should (1) display the default text and (2) display the qualifier text it builds from the associated X'82' subfield(s). It may format these text elements in any way that it chooses, so as to communicate to the operator that the display contains both default text and qualifier text.
- There are some causes and Recommended Actions which in some cases have qualifier text associated with them and in other cases do not. Given the scheme whereby the third digit of each code point indicates the number of X'82' subfields associated with the code point, there are two ways to allow for this situation:
  - Have multiple entries in Alert receivers' tables that index the same text, but have different numbers of X'82' subfields associated with them. (For example, have code points X'5501' and X'55A1' indexing the same text, but requiring, respectively, 0 and 1 X'82' subfields.)
  - Have a single entry for each text string, with the code point for it corresponding to the *maximum* number of strings of qualifier text associated with it. When a sender has no qualifier text strings to pass, or less than the full number of these strings, it sends "empty" X'82' subfields, i.e., subfields consisting only of a length byte (= X'02') and a key byte (= X'82').

  Since the first of these solutions would result in needless duplication in the Alert receivers' tables, the second has been chosen. There are thus two formats for the X'82' subfield, as indicated in Figure 10-14 on page 10-40.

"Empty" X'82' Subfield:



"Full" X'82' Subfield:

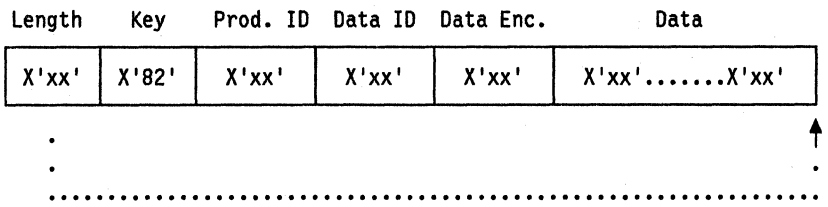


Figure 10-14. Structure of 'Empty' and 'Full' X'82' Subfields

The Qualified Message Data (X'01') Subfield

The Qualified Message Data (X'01') subfield, within the Detailed Data (X'98') subvector, transports a message code that indexes a message stored at the Alert receiver. National language support is the primary reason for transporting the message code rather than the message itself.

In some cases it is necessary to transport not only a message code, but also *qualifiers*, substitution text that is inserted into the message text indexed by the code. These qualifiers are themselves transported in Detailed Data (X'82') subfields that are pointed to from the Qualified Message Data subfield.

In many ways the Qualified Message Data subfield is similar to the Detailed Data (X'82') subfield, discussed in detail in "The Detailed Data (X'82') Common Subfield" on page 10-31. It carries a product ID code, a data ID (indicating the type of the message code), an indication of how the message code is itself encoded, and the message code. There are, however, two important differences between the two subfields:

- Whereas the detailed data in the X'82' subfield is itself displayed by the Alert receiver, the message code in the Qualified Message Data subfield *indexes* a stored message that is displayed.
- The Qualified Message Data subfield contains a Qualifier Count, indicating how many succeeding X'82' subfields are to be associated with the qualified message.

Figure 10-15 on page 10-41 illustrates the second of these points. The number shown in each Qualified Message Data (X'01') subfield is the value in its Qualifier Count field. This number indicates how many X'82' subfields are to be

processed in association with the X'01' subfield. In the upper example this number is 2, indicating that the next two X'82' subfields are to be treated as qualifiers of the message indexed by the X'01' subfield. The remaining two X'82' subfields are then treated as independent units of detailed data. In the lower example the Qualifier Count field indicates that only one X'82' subfield belongs with the message text. It is the responsibility of the Alert sender to insure that the proper number of Detailed Data subfields are associated each Qualified Message Data subfield it sends.

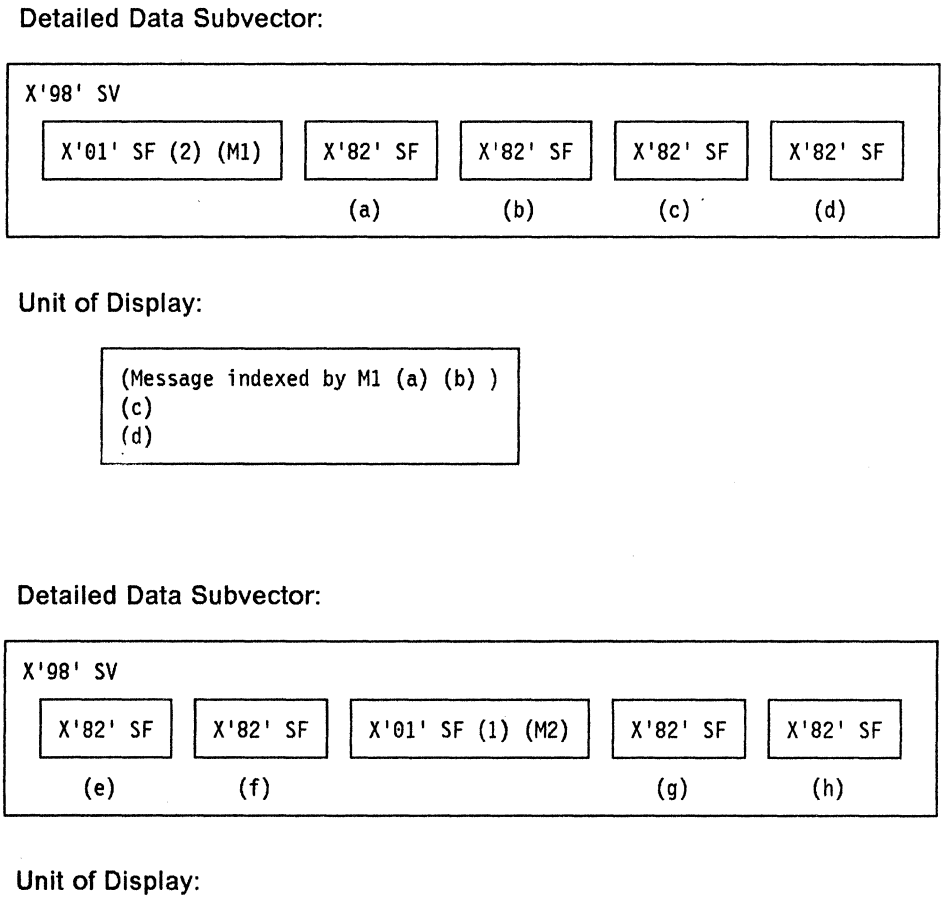


Figure 10-15. Relationship Between Qualified Message Data and Detailed Data Subfields

The text indexed by the message codes transported in the Qualified Message Data subfield is not architected in the way that the text indexed by the default/replacement Alert code points is. The architecture provides a mech-

anism that allows an Alert sender to specify to an Alert receiver which message is to be displayed; the messages themselves, however, are not defined by the architecture.

Support of the Qualified Message Data subfield is optional for an Alert receiver. When a particular Alert receiver does not support this subfield, the additional information contained in the message specified by the Alert sender is not available at that Alert receiver.

Hierarchy Information in Alerts

Hierarchy information in Alerts is reported in the Hierarchy/Resource List (X'05') MS common subvector. Even though this is a common subvector, appearing in a number of management services major vectors, the majority of its features are used only in Alerts. Thus the discussion of this subvector and the hierarchy information that it conveys appears here, in the Alerts section.

The Hierarchy Name List (X'10') Subfield:

The Hierarchy Name List (X'10') subfield provides logical identification of the resource being reported on by an Alert. It lists a hierarchy of resources, physically connected to each other, terminating in the Alert's indicated resource. For each resource, two pieces of information are provided: a name of up to eight characters, and a code point indicating the resource type.

An Alert receiver typically uses the hierarchy information supplied to it in two ways: it is required to present the complete hierarchy passed to it, but it may also, optionally, identify the indicated resource's name and type individually on a dynamic display of incoming Alerts. If an Alert receiver does elect to display the indicated resource individually, there is an additional piece of data from the X'10' subfield that it uses. For each resource identified in the X'10' subfield, the Alert sender specifies whether to display that resource as the Alert's indicated resource. The *display resource name indicator* is used as follows in constructing the individual display for an indicated resource:

- The resource *type* displayed is always the one associated with the last entry in the X'10' subfield.
- To choose a resource *name*, the Alert receiver starts at the end of the X'10' subfield and scans backwards. It skips over any entries for which the display resource name indicator is set to "do not display," scanning until it finds an entry for which the indicator says "display."
- If the resource name that it displays is not that from the last entry in the X'10' subfield, then the Alert receiver indicates on the screen in some way (e.g., by a special character) that the resource type and resource name it is displaying belong to different resources.

The goal of the display resource name indicator is to insure that the identification of the indicated resource is always meaningful to the operator at the Alert receiver. If the Alert receiver always displayed the name and type from the last entry in the X'10' subfield, this would not be the case.

An Alert from an SNA node reporting on a locally-attached non-SNA printer might, for example, have for the last entry in the X'10' subfield: name="PRINTER1," type=X'13' (printer). The name "PRINTER1" here is a local name, having meaning only at its own node; its role is to distinguish one of two local printers from the other one, PRINTER2. Every node in the network of the same type as the Alert sender might very well have its own PRINTER1 and PRINTER2.

In this situation, the Alert sender would set the display resource name indicator for the printer to "do not display"; the indicator for the sender itself would be set to "display." The operator at the Alert receiver would then see not "name=PRINTER1 type=PRINTER," which could have come from any of several nodes in the network, but instead "name=(SENDER) type=PRINTER *," where the asterisk indicates that (SENDER) and PRINTER refer to different resources. The operator thus knows which node has sent the Alert, and also that this node has experienced a problem with one of its printers. The full display of the hierarchy will indicate that the Alert is reporting on PRINTER1 rather than PRINTER2.

Correlation in Alerts

The Alert architecture provides an Alert sender with a means of correlating an Alert with *supporting data* about the problem, i.e., data about the problem stored locally by the Alert sender. The Supporting Data Correlation (X'48') common subvector transports one or more subfields identifying the supporting data associated with an Alert. If, for example, additional data related to a problem has been stored in a log at the Alert sender, the X'48' subvector can transport Detailed Data (X'82') subfields identifying both the file containing the supporting data and one or more search keys for the relevant record(s) in this file. The network operator is then in a position to correlate the Alert with the Alert sender's log entries for the same problem.

Examples of Physical and Logical Identification of the Origin of an Alert Condition

The following examples describe how the origin of an Alert condition (the failing network component) is identified in the Alert. This identification falls into two categories: *physical*, which flows in the Product Set ID (X'10') subvector, and *logical*, which may involve either or both of the SNA Address List (X'04') and Hierarchy/Resource List (X'05') subvectors.

In addition to data identifying the failing component, an Alert must also carry data identifying the PU sending the Alert.

Figure 10-16, Figure 10-17, and Figure 10-18 show the structure of the Product Set ID (PSID) subvectors that identify these resources.

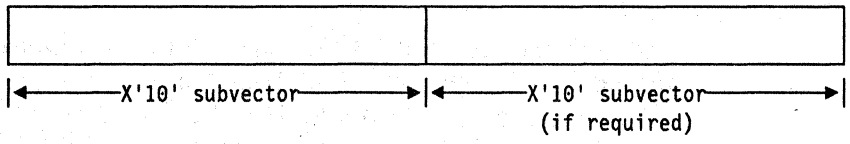


Figure 10-16. Product Set ID (X'10') Subvectors. One X'10' subvector contains data identifying the reporting PUMS. A second X'10' contains data identifying the problem origin. If the problem origin *is* the reporting PUMS, only the first X'10' is generated.

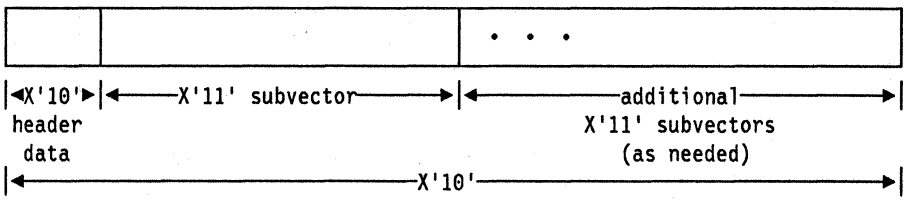


Figure 10-17. Product Identifier (X'11') Subvectors within a X'10' Subvector. A X'11' subvector is generated for each product (hardware or software) that makes up the entire product set to be identified.

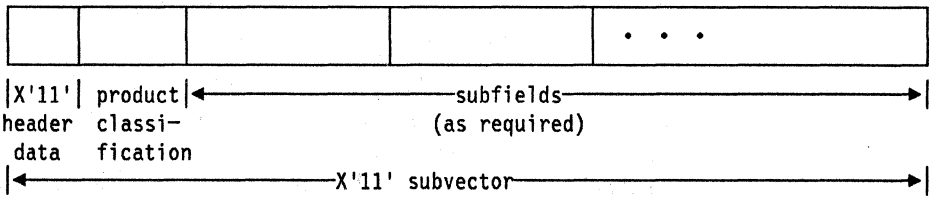


Figure 10-18. Product Identifier (X'11') Subvector. A X'11' subvector contains data to identify one hardware or software product. The product classification field identifies the product as hardware or software, and IBM or non-IBM. See *Systems Network Architecture Formats*, GA27-3136, for a description of which subfields are required to be present.

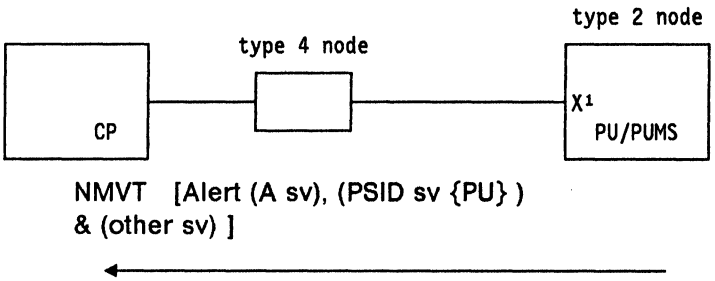
The examples in this section show all the subvectors that are required in an NMVT Alert, not just those provided by the EP_ALERT base subset.

The following figures illustrate some possible origins of Alert conditions (denoted X^1, X^2, X^3, X^4, X^5) in a network. The figures show only the portion of the network affected by the Alert condition and the resulting NMVT Alert flow. Below each configuration is an illustration of the NMVT Alert that flows on the SSCP-PU session. This NMVT is broken down to show explicitly the subvectors that contain the physical and logical identification of the components being identified.

EXAMPLE 1

Consider the type 2 node in Figure 10-19. If a temporary error is detected in the adapter for the link to the type 4 node (see error "X^1"), the PU (reporting PUMS) will generate an Alert. This Alert does not contain an SNA Address List subvector, since the PU name can be determined from the origin of the RU. The absence of an SNA Address List subvector identifies the PU itself as the origin of the Alert condition.

A single Product Set ID (X'10') subvector will contain data identifying the entity that is both the reporting PUMS and the problem origin.



KEY:

[ ] = contained in NMVT
( ) = subvectors contained in Alert major vector
{ } = component identified by a subvector

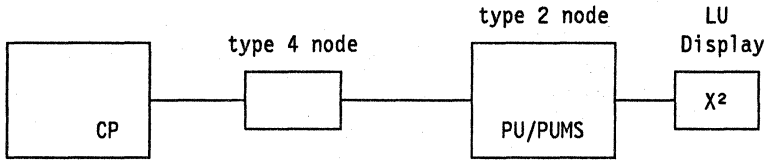
X^1:	denotes a failure within a device implementing the PU (e.g., temporary error in the adapter for the link to the type 4 node)
NMVT:	Network Management Vector Transport
Alert:	Alert major vector (X'0000')
A sv:	Alert subvectors (X'92'-X'98')
PSID sv:	Product Set ID subvector (X'10')
other sv:	other required (e.g., Date/Time (X'01') or Relative Time (X'42')) and optional (e.g., Self-Defining Text Message (X'31')) subvectors

Figure 10-19. Illustration of Example 1

EXAMPLE 2

Next consider the display in Figure 10-20 on page 10-46. If a failure is detected in the display itself (see error "X^2"), the LU LMS provides alert information to PUMS, so that PUMS can send an alert. This Alert will contain the SNA Address List subvector because the SNA component that is the problem origin is the LU.

One Product Set ID subvector identifies the reporting PUMS. A second Product Set ID subvector identifies the problem origin, the LU.



NMVT [Alert (SNAAL sv {l/a of LU}),
(A sv), (PSID sv { PU }),
(PSID sv { Display}), (other sv)]

KEY:

[ ] = contained in NMVT
( ) = subvectors contained in Alert major vector
{ } = component identified by a subvector

X²:	denotes a failure within a device implementing the LU or part of the LU function (e.g., hardware card failure in the display)
NMVT:	Network Management Vector Transport
Alert:	Alert major vector (X'0000')
A sv:	Alert subvectors (X'92'-X'98')
PSID sv:	Product Set ID subvector (X'10')
SNAAL sv:	SNA Address List subvector (X'04')
other sv:	other required (e.g., Date/Time (X'01') or Relative Time (X'42')) and optional (e.g., Self-Defining Text Message (X'31')) subvectors
l/a:	local address

Figure 10-20. Illustration of Example 2

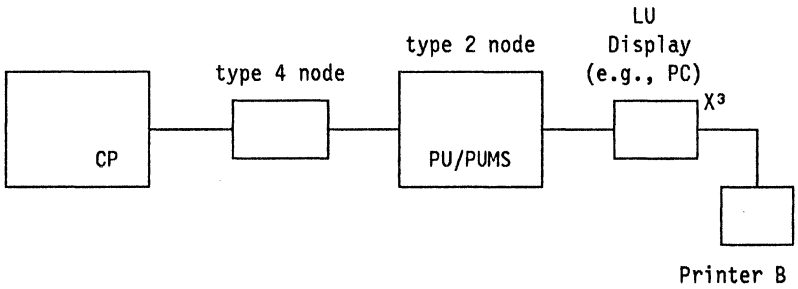
EXAMPLE 3

Consider the printer (printer B) attached to the display in Figure 10-21 on page 10-47. If a failure is detected that prohibits the LU from communicating with the printer (see error "X³"), the LU provides alert information so that PUMS can send an Alert. This Alert contains the SNA Address List subvector because the SNA component closest to the failure implements (at least part of) the LU. This Alert also contains the Hierarchy Name List subvector to identify the non-SNA addressable resource that is the origin of the Alert condition (the printer).

The Alert NMVT supports the presence of both the Hierarchy Name List and SNA Address List subvectors. These two subvectors will flow together on the SSCP-PU flow when a non-SNA addressable resource, controlled by a network device that is implementing an LU, fails. The SNA Address List subvector provides part of the SNA hierarchy by identifying the LU while the Hierarchy Name

List subvector provides the hierarchy from the LU to the failing non-SNA addressable resource.

A Product Set ID subvector will be generated that will describe the product set that contains the PUMS. The second PSID will identify the problem origin. In this case the machine is printer B.



NMVT [Alert (SNAAL sv {l/a of LU})
(A sv), (PSID sv {PU}),
(PSID sv {printer B}),
(HRL sv {Printer B})
(other sv)]

KEY:

[ ] = contained in NMVT
( ) = subvectors contained in Alert major vector
{ } = component identified by a subvector

X^3:	denotes a communication failure with the printer that is attached to a device implementing the LU or part of the LU function (e.g., power failure)
NMVT:	Network Management Vector Transport
Alert:	Alert major vector (X'0000')
A sv:	Alert subvectors (X'92'-X'98')
PSID sv:	Product Set ID subvector (X'10')
SNAAL sv:	SNA Address List subvector (X'04')
HRL sv:	Hierarchy/Resource List subvector (X'05')
other sv:	other required (e.g., Date/Time (X'01') or Relative Time (X'42')) and optional (e.g., Self-Defining Text Message (X'31')) subvectors
l/a:	local address

Figure 10-21. Illustration of Example 3

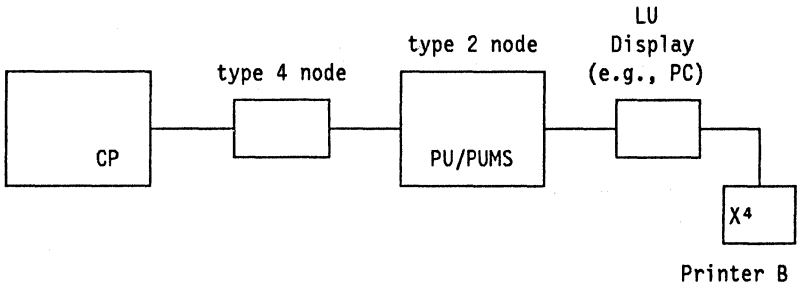
EXAMPLE 4

Consider the printer (printer B) attached to the display in Figure 10-22 on page 10-49. When a failure is detected by the RAS package within the printer (see error "X^4") the LMS for the printer provides alert information so that PUMS

can send an Alert. This Alert contains the SNA Address List subvector because the SNA component closest to the failure implements (at least part of) the LU. This Alert also contains the Hierarchy Name List subvector to identify the non-SNA addressable resource that is the origin of the alertable condition (the printer).

The Alert NMVT supports the presence of both the Hierarchy Name List and SNA Address List subvectors. These two subvectors will flow together, on the SSCP-PU flow, when a non-SNA addressable resource, controlled by a network device that is implementing the LU, fails. The SNA Address List subvector provides the SNA hierarchy by identifying the LU while the Hierarchy Name List subvector provides the hierarchy from the LU to the failing non-SNA addressable resource.

A PSID will be generated that will describe the product set that contains the PUMS. A second PSID will identify the problem origin, the printer attached to the LU. There will be no PSID associated with the LU since both the failing component and the PUMS have been positively identified.



NMVT [Alert (SNAAL sv {l/a of LU})
(A sv), (PSID sv {PU}),
(PSID sv {Printer B})
(HRL sv {Printer B})
(other sv)]

KEY:

[ ] = contained in NMVT
( ) = subvectors contained in Alert major vector
{ } = component identified by a subvector

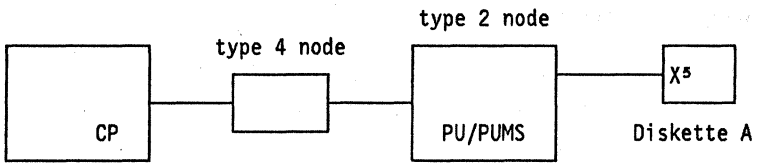
X[4]:	denotes a failure in the printer that is attached to a device implementing the LU or part of the LU function
NMVT:	Network Management Vector Transport
Alert:	Alert major vector (X'0000')
A sv:	Alert subvectors (X'92'-X'98')
PSID sv:	Product Set ID subvector (X'10')
SNAAL sv:	SNA Address List subvector (X'04')
HRL sv:	Hierarchy/Resource List subvector (X'05')
other sv:	other required (e.g., Date/Time (X'01') or Relative Time (X'42')) and optional (e.g., Self-Defining Text Message (X'31')) subvectors
l/a:	local address

Figure 10-22. Illustration of Example 4

EXAMPLE 5

Consider the diskette device (diskette A) attached to the type 2.0 node in Figure 10-23 on page 10-50. If a failure is detected in the diskette drive (see error "X[5]"), the LMS provides alert information to PUMS so that PUMS can send an Alert. This Alert contains the Hierarchy Name List subvector to identify the non-SNA addressable resource that is the origin of the Alert condition. This will provide the hierarchy from the PU to the failing resource. A maximum of five entries (an entry = a resource name + resource type index) are allowed in a Hierarchy Name List subvector.

A PSID will identify the reporting PUMS. A second PSID will be generated to identify the problem origin, the diskette.



NMVT [Alert (A sv),
(HRL sv {Diskette A}),
(PSID sv {PU}),
(PSID sv {Diskette A}),
& (other sv) [

KEY:

[ ] = contained in NMVT
( ) = subvectors contained in Alert major vector
{ } = component identified by a subvector

X^5:	denotes a failure in the diskette or diskette drive that is attached to the device implementing the PU/PUMS
NMVT:	Network Management Vector Transport
Alert:	Alert major vector (X'0000')
A sv:	Alert subvectors (X'92'-X'98')
PSID sv:	Product Set ID subvector (X'10')
HRL sv:	Hierarchy/Resource List subvector (X'05')
other sv:	other required (e.g., Date/Time (X'01') or Relative Time (X'42')) and optional (e.g., Self-Defining Text Message (X'31')) subvectors

Figure 10-23. Illustration of Example 5

EP_RTM Function Set

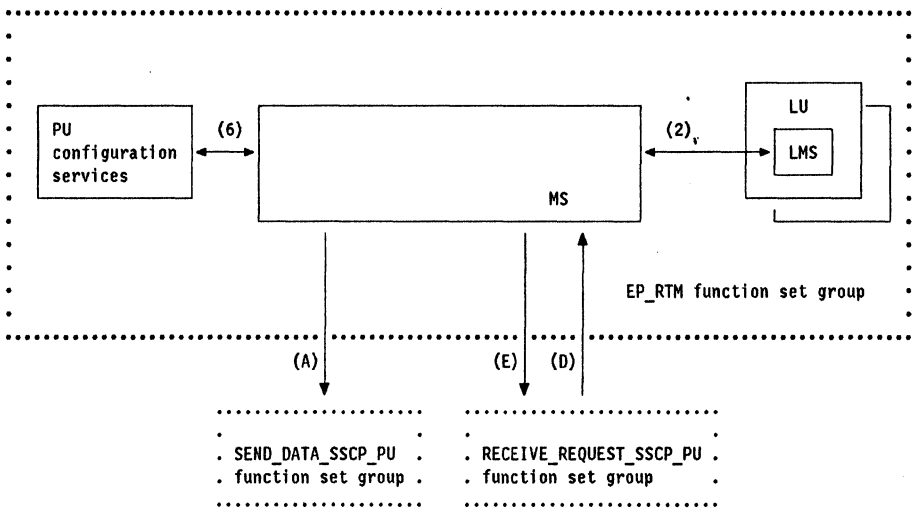


Figure 10-24. EP_RTM Function Set Group

The EP_RTM function set provides the capability to support Response-Time Monitoring (RTM). This is defined as the capability to quantify, measure, and report end-user response times for LUs controlled by this PU according to definitions specified on NMVT requests.

Refer to Figure 10-24 throughout the discussion of the EP_RTM function set.

Protocol Boundaries with Components Outside EP_RTM

- Input:
  - Internal MS protocol boundary D (NMVT Received)

    The EP_RTM function set group receives requests from the RECEIVE_REQUEST_SSCP_PU function set group to process incoming NMVTs. It is passed the NMVT and the name of the CP from which it was received. The details of this protocol boundary are described in "Protocol Boundary D - NMVT Received" on page 8-19.

- Output:
  - Internal MS protocol boundary A (Send NMVT)

    The EP_RTM function set group requests the SEND_DATA_SSCP_PU function set group to send an NMVT RU on the SSCP-PU session with its controlling CP. The output data consists of the complete NMVT and the address of the control point to which it is to be sent. The details of this protocol boundary are described in "Protocol Boundary A - Send NMVT" on page 8-18.

  - Internal MS protocol boundary E (Send NMVT Response)

    The EP_RTM function set group requests the PUMS_RECEIVE_NMVT function set group to send an NMVT response RU on an SSCP-PU session with a

specified CP. The output data consists of the CP address and sense data. The details of this protocol boundary are described in "Protocol Boundary E - Send NMVT Response" on page 8-19.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

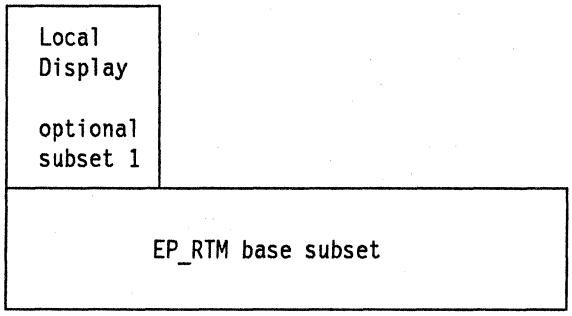


Figure 10-25. Base and Optional Subsets of EP_RTM Function Set

EP_RTM Base Subset

Functions Provided

The EP_RTM base subset provides the capability to monitor end-user response times as described by "Response-Time Monitoring (RTM)" on page 4-3.

Formats Supported

The EP_RTM base subset supports the receiving of an NMVT containing a Request Response-Time Monitor (X'8080') major vector.

The EP_RTM base subset supports the sending of an NMVT containing a Response Time Monitor (X'0080') major vector. The following subvectors are built by EP_RTM:

- A Date/Time (X'01') or Relative Time (X'42') MS common subvector
- An SNA Address List (X'04') MS common subvector
  - When sending unsolicited data and status
  - When positively replying to a request for data and status
- A Data Reset Flag (X'45') MS common subvector
  - When a set of RTM counters has been reset
- A Sense Data (X'7D') MS common subvector

  - When a request could not be successfully completed, and the elective for reporting this condition in a reply has been selected
- An RTM Status Reply (X'91') subvector
  - When sending unsolicited data and status
  - When positively replying to a request for data and status
- An RTM Data (X'93') subvector
  - When sending unsolicited data and status
  - When positively replying to a request for data and status for an LU that has accumulated data

Electives

There are two electives available to this base subset:

1. The reporting of requests that cannot be processed by sending sense data in either a negative response or a reply.
2. Always sending a reply for the last configured LU after a request for RTM data and status for all LUs with accumulated data. This elective allows the PU to send replies for earlier LUs immediately, without having to hold each one until it can be determined whether it is the last reply for the request.

Implementation Requirements

The requirements for implementing the EP_RTM base subset are described by a model consisting of subsets of PU configuration services, an LU LMS, and PU management services.

A Subset of PU Configuration Services:

- Interacts with PUMS via MS protocol boundary 6 as follows:
  - Provides a list of all LUs configured at the node when requested by PUMS

A Subset of an LU LMS:

- Interacts with PUMS via MS protocol boundary 2 as follows:
  - Sets RTM parameters when requested by PUMS. If the LU LMS was not successful, the reason the request could not be completed is passed to PUMS.
  - Provides RTM status and data when requested by PUMS.

    The LU LMS transfers the requested RTM data and status to PUMS. If the request could not be acted upon, the reason is passed to PUMS.
  - If RTM parameters so indicate, provides RTM data and status to PUMS whenever an RTM counter overflows, or whenever a session is deactivated
- Manages the collection of RTM data as follows:
  - Activates the RTM function whenever PUMS requests that the RTM parameters be set

Default parameters are used for any values not specified.

- Deactivates the RTM function as follows:
  - When the node is IPLed
  - When an ACTPU (cold) is received
  - When requested by PUMS
- Resets RTM counters for its specific LU as follows:
  - When the node is IPLed
  - When an ACTPU (cold) is received
  - When an ACTLU is received
  - Whenever the counters for that LU are sent unsolicited
  - When RTM is deactivated by a request from PUMS
  - When a reset is requested by PUMS
  - When a change to the RTM parameters is requested by PUMS

A Subset of PU Management Services:

- Provides the following processes:
  - Receiving RTM requests
    - Upon request by the RECEIVE_REQUEST_SSCP_PU function set group (via protocol boundary D), does the following:
      - Parses the requests for validity and requests the RECEIVE_REQUEST_SSCP_PU function set group (via protocol boundary E), to send a positive or negative response.
      - If the request parsed correctly and it was for a single LU, processes the request by requesting the specified LU to set RTM measurement parameters or gather RTM data.
      - If the request parsed correctly and it was for all LUs, processes the request by requesting all LUs to set RTM parameters or to gather RTM data.
  - Sending RTM data

    Upon request by a subset of an LU LMS, does the following:
    - Receives RTM data and status, or status only, from the LU
    - Receives sense data from the LU

    After receiving the data, it builds an NMVT containing one or more of the subvectors described in "Formats Supported," except in the following cases involving replies to a request for RTM data and status from all LUs with accumulated data:
    - The LU has provided status only, and it is not the last configured LU
    - The LU has provided status only, it is the last configured LU, both data and status have been returned for the current request by at

least one earlier LU, and Elective 2 (always reply for the last configured LU) has not been implemented

Note that an LU can provide status without accompanying data only in reply to a request for RTM data and status; when RTM information comes from the LU unsolicited, it always includes both data and status. Note also that a reply is always sent to a request for RTM data and status from a specified LU, even if the LU returns status only, regardless of whether the LU is the PU's last configured LU.

After this the NMVT is passed to the SEND_DATA_SSCP_PU function set group via protocol boundary A.

Receiving RTM Requests:

This process is started by the process described in "Receiving NMVTs" on page 9-18 (RECEIVE_REQUEST_SSCP_PU function set group), when the latter has identified the major vector key X'8080' (Request Response Time Monitor) in a request NMVT. The first task of this process is to parse the RTM major vector. If there is a syntactic error in the major vector, or in one of the subvectors it contains, or if PUMS in the node fails to support some function presupposed by the request, then this process starts the process described in "Sending NMVT Responses" on page 9-19 to send the appropriate -RSP. The sense data that can be sent by PUMS at this time are detailed in Table 10-38 on page 10-156. If no condition requiring a -RSP is found, then this process starts the process described in "Sending NMVT Responses" (via protocol boundary E) to send a +RSP.

Table 10-4. Subvector Conditions of Presence in the Request RTM (X'8080') Major Vector

Request Type	**Subvectors present in the Request RTM (X'8080') major vector**
Retrieve data for all LUs with accumulated data	X'92'
Retrieve data for a specified LU	X'92', X'04'
Set parameters for all LUs	X'92', X'94'
Set parameters for a specified LU	X'92', X'94', X'04'

The Receiving RTM Requests process handles each of the four types of requests that it may receive differently. Table 10-4 lists the four request types and the subvectors that are present in each. Summarizing the information in Table 10-4:

- The RTM Request (X'92') subvector is present on all requests.
- The SNA Address List (X'04') subvector is present on requests directed to a single LU, and absent from requests directed to all of a PU's LUs.
- The RTM Control (X'94') subvector is present on requests to set RTM parameters, and absent from requests for RTM data.

After determining the type of request it has received, and verifying that the request is syntactically correct, the Receiving RTM Requests process proceeds as follows:

- *Set RTM parameters for a specified LU:*

  Since this request expects no reply, no request/reply correlation is needed. This process sends a request to the specified LU's LMS to set its RTM parameters to the values specified in the request. Note that one of these parameters specifies whether the RTM function itself is to be active for the LU.

  Each LU LMS retains a single set of RTM parameter settings to use. It receives a default set at system initialization time, and subsequently overwrites them as it receives parameter setting requests from the Receiving RTM Requests process. When the PU is IPLed or receives an ACTPU (cold), the default parameters supplied at system generation time are restored for all LUs.

  Two points should be noted here:

  - Since a request to set RTM parameters expects no reply, all exception conditions must be reported on -RSPs. In particular, the check to see whether the LU local address specified in the request is known to PU configuration services must be done prior to the decision to send a +RSP or a -RSP.
  - PU configuration services has knowledge of, and this process is able to communicate with, all LUs configured at the node, not just those that are currently active. If the LU specified in this request is currently inactive, its LMS will accept and save the RTM parameters passed to it on the request, for use whenever the LU does become active.

  After passing the RTM parameter settings to the specified LU, this process terminates.

- *Set RTM parameters for all LUs:*

  Since this request expects no reply, no request/reply correlation is needed. This process asks PU configuration services for a list of all the LUs configured at this node. It then sends a request to each LU LMS to set its RTM parameters to the values specified in the request. An LMS for an LU that is currently inactive saves the values passed to it, for use when the LU does become active.

  After passing the RTM parameter settings to all the LUs, this process terminates.

- *Retrieve RTM data and status for a specified LU:*

  This process first builds an entry for the request in its correlation control block. Figure 10-26 on page 10-57 shows the structure of this control block entry. The following values are supplied by this process.

  - The control point address that was passed to this process from the RECEIVE_REQUEST_SSCP_PU function set group.

    This address is received via MS protocol boundary D item 2.

- The value that came in the PRID field (bits 4–15 of bytes 5–6) of the request NMVT
- A list containing, as its only entry, the LU local address that came in the SNA Address List (X'04') subvector in the request
- A null entry for the NMVT buffer

After creating the control block entry, this process passes to the LMS of the specified LU the request for RTM data. This process then terminates.

- *Retrieve RTM data and status for all LUs that have accumulated data:*

  This process first builds an entry for the request in its correlation control block. Figure 10-26 shows the structure of this control block entry. The following values are supplied by this process.

  - The control point address that was passed to this process from the RECEIVE_REQUEST_SSCP_PU function set group.

    This address is received via MS protocol boundary D item 2.
  - The value that came in the PRID field (bits 4–15 of bytes 5–6) of the request NMVT
  - A list containing the local addresses of all the LUs configured at the node; this process obtained the list by asking PU configuration services for it
  - A null entry for the NMVT buffer

  After creating the control block entry, this process passes to each LU LMS in the node a request for RTM data. This process then terminates.

Table 10-38 on page 10-156 lists the sense data that this process can pass to the RECEIVE_REQUEST_SSCP_PU function set group via internal MS protocol boundary E item 2. The address of the CP that sent the request is passed in item 1. The value X'0000 0000' is used to indicate a +RSP. The remaining values, representing the various errors that this process may detect in an RTM request, are to be sent in a -RSP. Sense data X'0870 nn00' is sent for an invalid value in an unformatted subvector. For example, if the RTM measurement definition X'04'is not supported, sense data X'0870 9408' is used (byte 8 of the X'94' subvector is in error).

The NMVT Buffer:

CP ADDR	PRID	LU ADDRESS LIST	NMVT BUFFER

Figure 10-26. EP_RTM's Request/Reply Correlation Control Block

The discussion of the Sending RTM Data process must recognize that there are differences in the process depending upon whether EP_RTM has implemented Elective 2. Recall that Elective 2 allows an implementation of EP_RTM to avoid the overhead associated with lookahead by always sending a reply for its last configured LU in reply to a request for data and status for all LUs with accumu-

lated data, even if that LU has returned only status information. An instance of EP_RTM not choosing to implement Elective 2 does not send a reply in this case, so long as at least one of the preceding LUs has returned both data and status. In summary, Elective 2 eliminates the overhead associated with lookahead, but at the cost of an NMVT flow that is not really necessary.

In order to perform the necessary lookahead, an instance of EP_RTM not opting to implement Elective 2 must make use of an additional column in its correlation control block: the *NMVT buffer*. This buffer holds one NMVT that was built earlier by the Sending RTM Data process. Its purpose is to give the Sending RTM Data process a chance to change the setting of the sequence field in the NMVT header of the buffered NMVT if it turns out that no further replies are forthcoming for the request. A simple example illustrates how this buffer works; see Table 10-5 on page 10-59. Note that in this example the first two columns of the correlation control block entry are not shown, since they do not vary.

Table 10-5. Example Illustrating Use of EP_RTM's NMVT Buffer. The example begins with a request for RTM data and status from all LUs with accumulated data, directed to a node at which the four LUs A, B, C, and D are defined.

Sent to CPMS	LU Addresses	Contents of the NMVT Buffer
• *Receiving NMVTs and Receiving RTM Requests processes receive and process the request* • *Correlation control block entry for the request is created* • *Requests for RTM data and status are issued to LUs A, B, C, and D*		
(nothing)	A, B, C, D	(empty)
• *LU A returns data and status* • *Sending RTM Data process builds NMVT for A* • *Sending RTM Data process places the NMVT for A in the NMVT buffer*		
(nothing)	B, C, D	NMVT for A (seq. = first)
• *LU B returns data and status* • *Sending RTM Data process builds NMVT for B* • *Sending RTM Data process removes the NMVT for A from the NMVT buffer and passes it to the Sending NMVTs process; since another NMVT (for LU B) will also be sent in reply to the request, the sequence setting "first" (rather than "only") is correct.* • *Sending NMVTs process sends the NMVT for A to CPMS* • *Sending RTM Data process places the NMVT for B in the RTM buffer*		
NMVT for A (seq. = first)	C, D	NMVT for B (seq. = middle)
• *LU C returns status only* • *Sending RTM Data process removes C from the correlation control block entry*		
(nothing)	D	NMVT for B (seq. = middle)
• *LU D returns status only* • *Sending RTM Data process removes D from the correlation control block entry* • *Sending RTM Data process removes the NMVT for B from the RTM buffer, adjusts its sequence field (from "middle" to "last"), and passes it to the Sending NMVTs process* • *Sending NMVTs process sends the NMVT for B to CPMS*		
NMVT for B (seq. = "last")	(empty)	(empty)

Sending RTM Data:

This process is started as follows:

- By an LU LMS for the following conditions:
  - To report RTM data and status, or RTM status only if the LU has no accumulated data
  - To report that a request could not be completed

This will occur only when a request could not be completed and the elective for reporting this condition on a reply has been selected.

The function of this process is to construct an NMVT containing the subvectors described in "Formats Supported"; Table 10-6 shows which subvectors are constructed in which cases.

After building the NMVT, this process starts the process described in "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set group) via protocol boundary A to send the NMVT to CPMS.

Table 10-6. Subvector Conditions of Presence in the RTM (X'0080') Major Vector

Reply Type	**Subvectors present in the RTM (X'0080') major vector**
Positive reply to a request for RTM data and status from an LU with accumulated data	X'91', X'93', X'04'
Positive reply to a request for RTM data and status from an LU with no accumulated data	X'91', X'04'
Negative reply to a request for RTM data and status, if the elective for reporting LU-detected exception conditions in replies has been selected	X'7D', X'04'

Recall the earlier comment in "The NMVT Buffer" on page 10-57, that the details of how RTM data is sent to CPMS vary depending on whether Elective 2 has been implemented. Consequently, there are two descriptions of how this data is sent:

Sending Replies If Elective 2 is Implemented:

Except in a single case described below, this process builds an NMVT when it receives RTM data, RTM status, or an SNA sense data from an LU LMS.

- If this process was passed both RTM data and status by the LU LMS when it was started, it builds both the RTM Status Reply (X'91') and RTM Data (X'93') subvectors.
- If this process was passed an SNA sense data by the LU LMS, it builds a Sense Data (X'7D') subvector.
- If this process was passed RTM status without any accompanying data, it must examine its correlation control block to determine what action to take:
  - If the control block entry for this PRID contains other local addresses besides that of the LU that invoked this process, then it does not build an NMVT, and does not start the process described in "Sending NMVTs" on page 9-14 to send a reply. Its only action is to delete the invoking LU's local address from the control block entry.

    This state of the correlation control block entry occurs only for a request for RTM data and status from all LUs with accumulated data.

  - If the local address of the LU that invoked this process is the only one in the control block entry for this PRID, then this process builds a RTM Status Reply (X'91') subvector.

    This state of the correlation control block entry covers three cases:

    - The request was for RTM data and status for a single specified LU.
    - The request was for RTM data and status from all LUs with accumulated data, this status is coming from the last configured LU, and none of the preceding LUs has returned any data.
    - The request was for RTM data and status from all LUs with accumulated data, this status is coming from the last configured LU, and at least one preceding LU has returned both status and data.

- If it has built either the RTM Status Reply (X'91') or the RTM Data (X'93') subvector, this process also builds the SNA Address List (X'04') subvector; see "Building the SNA Address List (X'04') Subvector" on page 10-148 for details.
- If it is building an NMVT, then depending on the node's capabilities, this process builds either a Date/Time (X'01') or Relative Time (X'42') subvector; see "Building the Date/Time (X'01') and Relative Time (X'42') Subvectors" on page 10-148 for details.
- If it is building an NMVT, then if the data from the LU LMS indicates that RTM counters have been reset, this process builds a Data Reset Flag (X'45') subvector.
- After building all the appropriate subvectors, this process places them in an RTM (X'0080') major vector within an NMVT. See "Building a Management Services Major Vector" on page 10-151 and "Building an NMVT" on page 10-151 for details. The only constraint on the placement of the subvectors within the major vector is that if it is present, the SNA Address List subvector must appear first.
- After completing an NMVT, this process starts the process described in "Sending NMVTs" on page 9-14 to send the NMVT to CPMS."Protocol Boundary A - Send NMVT" on page 8-18 shows the items passed when starting that process. This process places the following values in these items:

  Item 1: A pointer to the NMVT that it has built

  Item 2: The address of the control point that sent the request

- After this process has started the "Sending NMVTs" process, it terminates.

Sending Replies If Elective 2 is Not Implemented:

Except in a single case, slightly different from that described earlier, this process builds an NMVT when it receives RTM data, RTM status, or an SNA sense data from an LU LMS.

- If this process was passed both RTM data and status by the LU LMS when it was started, it builds both the RTM Status Reply (X'91') and RTM Data (X'93') subvectors.
- If this process was passed an SNA sense data by the LU LMS, it builds a Sense Data (X'7D') subvector.
- If this process was passed RTM status without any accompanying data, it must examine the correlation control block to determine what action to take. Note that in this case the correlation control block has the fourth column, for the NMVT buffer.
  - If the control block entry for this PRID contains other local addresses besides that of the LU that invoked this process, then it does not build an NMVT. Its only action is to delete the invoking LU's local address from the control block entry.
  - If the local address of the LU that invoked this process is the only one in the control block entry for this PRID, then this process examines the NMVT buffer:
    - If the buffer is empty, then this process builds an RTM Status (X'91') subvector reporting the status of the LU that invoked it.
    - If the buffer is full, then this process does not build an NMVT for the LU that invoked it. Instead, it updates the sequence field of the NMVT in the buffer (from "first" to "only" or from "middle" to "last"). It then passes the NMVT to the process described in "Sending NMVTs" on page 9-14 via Protocol Boundary A.
- If it has built either the RTM Status Reply (X'91') or the RTM Data (X'93') subvector, this process also builds the SNA Address List (X'04') subvector; see "Building the SNA Address List (X'04') Subvector" on page 10-148 for details.
- If it is building an NMVT, then depending on the node's capabilities, this process builds either a Date/Time (X'01') or Relative Time (X'42') subvector; see "Building the Date/Time (X'01') and Relative Time (X'42') Subvectors" on page 10-148 for details.
- If it is building an NMVT, then if the data from the LU LMS indicates that RTM counters have been reset, this process builds a Data Reset Flag (X'45') subvector.
- After building all the appropriate subvectors, this process places them in an RTM (X'0080') major vector within an NMVT. See "Building a Management Services Major Vector" on page 10-151 and "Building an NMVT" on page 10-151 for details. The only constraint on the placement of the subvectors within the major vector is that if it is present, the SNA Address List subvector must appear first.
- Sending a reply. When Elective 2 is implemented, the decision when to send a reply is straightforward: any reply that is constructed is sent as soon as it is created. The decision is more complicated if EP_RTM does not implement Elective 2:

- Ordinarily, when a reply NMVT is created, it is placed in the NMVT buffer; if the buffer already contains an NMVT, that NMVT is passed to the Sending NMVTs process. When an NMVT is placed in the buffer, its sequence field is set to either "first" or "middle," depending upon whether a reply has already been sent for its request. When an NMVT is removed from the buffer and passed to the Sending NMVTs process because another NMVT is taking its place in the buffer, its sequence field is not changed.
- If the NMVT buffer is empty, and the Sending RTM Data process creates an NMVT for the last remaining LU in the correlation control block entry, then the NMVT is passed directly to the Sending NMVTs process. This NMVT's sequence field is set to "only."

Regardless of whether it is an NMVT it has just built, or one that has been in the NMVT buffer, the Sending RTM Data process starts the Sending NMVTs process in the same way. "Protocol Boundary A - Send NMVT" on page 8-18 shows the items passed when starting that process. The Sending RTM Data process places the following values in these items:

Item 1: A pointer to the NMVT that it has built

Item 2: The address of the control point that sent the request

After the Sending RTM Data process has started the Sending NMVTs process, it completes any of the processing described above that it has not finished: e.g., placing an additional NMVT in the NMVT buffer. Then it terminates.

Sending Unsolicited RTM Data:

Unsolicited RTM data is handled the same way by the Sending RTM Data process, regardless of whether it has implemented Elective 2. The only data that this process receives unsolicited from an LU LMS is RTM data and status for the LU. When it receives this data, the process builds the RTM Status Reply (X'91'), the RTM Data (X'93'), the SNA Address List (X'04'), the Data Reset Flag (X'45'), and either the Date/Time (X'01') or the Relative Time (X'42') subvectors; it places these subvectors in an RTM (X'0080') major vector within an NMVT. It then starts the Sending NMVTs process, and passes it the NMVT. "Protocol Boundary A - Send NMVT" on page 8-18 shows the items passed when the Sending NMVTs process is started. The Sending RTM Data process places the following values in these items:

Item 1: A pointer to the NMVT that it has built

Item 2: 'NONE' (This data is unsolicited.)

After the Sending RTM Data process has started the Sending NMVTs process, it terminates.

EP_RTM Optional Subset 1 (Local Display)

Functions Provided

EP_RTM optional subset 1 (Local Display) provides the capability to display RTM data at the node implementing this function set group, and to accept commands from the host to enable or disable the local display. The mechanism by which the data is displayed is implementation defined.

Formats Supported

This optional subset supports the processing of bit 7 (local display of RTM data) in the RTM status and control change mask bytes and RTM status and control indicators bytes of the RTM Control (X'94') subvector. It also supports the setting of bit 7 (local display of RTM data) in the RTM status byte of the RTM Status Reply (X'91') subvector. The setting of this bit is dependant on the status (enabled/disabled) of the local display function.

Implementation Requirements

The requirements for implementing the EP_RTM optional subset 1 (Local Display) are described by a model consisting of subsets of LU LMS and PU management services.

A Subset of LU LMS:

- Provides a facility for locally displaying RTM data
- Enables and disables the local display of RTM data when requested by PUMS

A Subset of PU Management Services:

- A subset of the "Receiving RTM Requests" process
  - Processes requests containing bit 7 (local display of RTM data) in the RTM status and control change mask bytes and RTM status and control indicator bytes of the RTM Control (X'94') subvector and transfers these requests to an LU LMS
- A subset of the "Sending RTM Data" process
  - Processes LMS data providing the status of local RTM display and transfers data into bit 7 (local display of RTM data) in the RTM status byte of the RTM Status Reply (X'91') subvector

EP_QPI Function Set

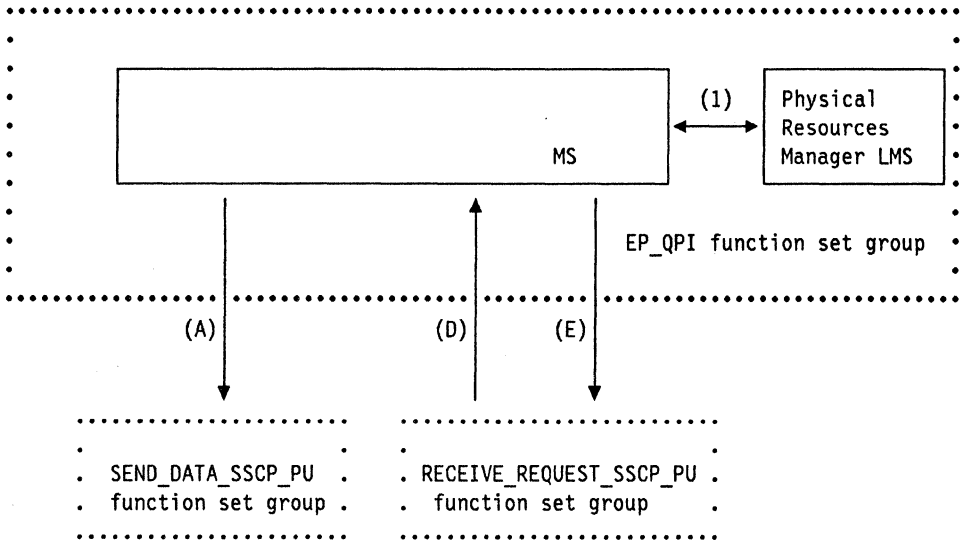


Figure 10-27. EP_QPI Function Set Group

The EP_QPI function set provides the ability to physically identify a node and optionally, its dependent resources upon request.

Refer to Figure 10-27 throughout the discussion of the EP_QPI function set.

Protocol Boundaries with Components Outside EP_QPI

- Input:
  - Internal MS protocol boundary D (NMVT Received)

    The EP_QPI function set group receives requests from the RECEIVE_REQUEST_SSCP_PU function set group to process incoming NMVTs. It is passed the NMVT and the name of the CP from which it was received. The details of this protocol boundary are described in "Protocol Boundary D - NMVT Received" on page 8-19.

- Output:
  - Internal MS protocol boundary A (Send NMVT)

    The EP_QPI function set group requests the SEND_DATA_SSCP_PU function set group to send an NMVT RU on the SSCP-PU session with its controlling CP. The output data consists of the complete NMVT and the address of the control point to which it is to be sent. The details of this protocol boundary are described in "Protocol Boundary A - Send NMVT" on page 8-18.

  - Internal MS protocol boundary E (Send NMVT Response)

    The EP_QPI function set group requests the RECEIVE_REQUEST_SSCP_PU function set group to send an NMVT response RU on an SSCP-PU session

with a specified CP. The output data consists of the CP address and sense data. The details of this protocol boundary are described in "Protocol Boundary E - Send NMVT Response" on page 8-19.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

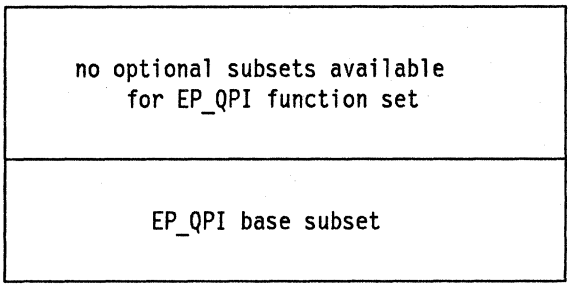


Figure 10-28. Base and Optional Subsets of EP_QPI Function Set

EP_QPI Base Subset

Functions Provided

The EP_QPI base subset provides the capability to physically identify the SNA node and port-attached devices upon request. For additional details refer to "Query Product ID (QPI)" on page 5-3.

Formats Supported

The EP_QPI base subset supports the receiving of an NMVT containing a Request Product Set ID (X'8090') major vector.

The EP_QPI base subset supports the sending of NMVTs containing a Reply Product Set ID (X'0090') major vector. The following subvectors are built by EP_QPI:

- A Date/Time (X'01') or Relative Time (X'42') MS common subvector. For details, see "Building the Date/Time (X'01') and Relative Time (X'42') Subvectors" on page 10-148.
- An SNA Address List (X'04') MS common subvector. For details, see "Building the SNA Address List (X'04') Subvector" on page 10-148.
- A Product Set ID (X'10') MS common subvector. For details, see "Building the Product Set ID (X'10') Subvector" on page 10-149.
- A Product Identifier (X'11') MS common subvector. For details, see "Building the Product Set ID (X'10') Subvector" on page 10-149.

- An Attached Device Configuration Description (X'82') MS subvector

Implementation Requirements

The requirements for implementing the EP_QPI base subset are described by a model consisting of subsets of management services and the Physical Resources Manager LMS.

A Subset of Physical Resources Manager LMS: This subset interacts with MS via protocol boundary 1 by providing identification of product sets. The Physical Resources Manager LMS provides the identification of the product set making up the SNA node. If requested by EP_QPI, the Physical Resources Manager LMS also provides identification of product sets in port-attached devices.

For the SNA node and (if requested) each port-attached devices, the Physical Resources Manager LMS replies with a product set identification message. The message contains:

- Sequencing information:
  - The only reply for its request
  - The first reply for its request
  - The last reply for its request
  - A middle reply for its request

  Note: This sequencing information requires the Physical Resources Manager LMS to "look-ahead" to know if there will be at least one more reply for the product set identification request.
- LU address
- Product information so EP_QPI can build the Product Set ID (X'10') MS common subvector (which includes the Product Identifier (X'11') MS common subvector) and Attached Device Configuration Description (X'82') MS subvector. For details of the X'10' subvector, see "Building the Product Set ID (X'10') Subvector" on page 10-149.

A Subset of Management Services: This subset provides the following processes:

- Receiving QPI requests

  Upon request by the RECEIVE_REQUEST_SSCP_PU function set group (via protocol boundary D), does the following:
  - Parses the request for validity and requests the RECEIVE_REQUEST_SSCP_PU function set group (via protocol boundary E) to send a positive or negative response.
  - If the request parsed correctly, processes the request by requesting the Physical Resources Manager LMS (via protocol boundary 1) to provide product set identification. The request from MS to the Physical Resources Manager LMS specifies one of the following:

  - Only product set identification of the SNA node is requested (i.e., the Request Product Set ID (X'8090') major vector contained a X'81' subvector)
  - Product set identification of the SNA node and identification of the product set in each port-attached device is requested (i.e., the Request Product Set ID (X'8090') major vector contained a X'83' subvector)

- Sending QPI data

  Upon request by a subset of the Physical Resources Manager LMS, does the following:

  - Receives (from the Physical Resources Manager LMS via MS application protocol boundary 1) one data message for the product set comprising the PU and (if requested) one data message for each port-attached product set. The Physical Resources Manager LMS also passes to MS the correlator that it received with the request.
  - Uses the correlator to retrieve the PRID from the request.
  - Using a data message received from the Physical Resources Manager LMS and the PRID it has retrieved, builds a complete NMVT.
  - Passes the NMVT to the SEND_DATA_SSCP_PU function set group via protocol boundary A.

Receiving QPI requests:

This process is started by the process described in "Receiving NMVTs" on page 9-18 (RECEIVE_REQUEST_SSCP_PU function set group), when the latter has identified the major vector key X'8090' (Request Product Set ID) in a request NMVT. The first task of this process is to parse the product identification major vector. If there is a syntactic error in the major vector, or in one of the subvectors it contains, or if PUMS in the node fails to support some function presupposed by the request, then this process starts the process described in "Sending NMVT Responses" on page 9-19 to send the appropriate -RSP. The sense data that can be sent by MS are detailed in "Parsing of NMVTs" on page 10-152 and Table 10-39 on page 10-156. If no condition requiring a -RSP is found, then this process starts the process described in "Sending NMVT Responses" (via protocol boundary E) to send a +RSP,

This process accepts either the X'81' or the X'83' subvector in the Request Product Set ID (X'8090') major vector.

This process also creates a correlator that it will pass to the Physical Resources Manager LMS with the request for product set identification. EP_QPI retains an association between this correlator and the control point name and PRID from the request. The purpose of this private correlator, used only on protocol boundary 1 is to shield the Physical Resources Manager LMS from differences in, and changes to, the manner in which management services data is transported between nodes. Since knowledge of request/reply correlation is used by transport components as well as by node components to communicate, correlators, such as the PRID, that actually flow between nodes tend to be

transport-specific. By mapping the PRID to its own private correlator, EP_QPI insures that the Physical Resources Manager LMS will not be affected by changes to the underlying management services transport.

This process now passes to the Physical Resources Manager LMS (via MS protocol boundary 1) the private correlator and a request for product set identification data. This process then terminates.

"Parsing of NMVTs" on page 10-152 and Table 10-39 on page 10-156 lists the sense data that this process can send to the RECEIVE_REQUEST_SSCP_PU function set group via internal MS protocol boundary E item 2. (See "Protocol Boundary E - Send NMVT Response" on page 8-19.). The address of the CP that sent the request is passed in item 1. The value X'0000 0000' is used to indicate a +RSP. The remaining values, representing the various syntactic errors that this process may detect in a query product identification request, are to be sent in a -RSP.

Sending QPI data:

This process is started by the Physical Resources Manager LMS when it has product set identification data to send. The Physical Resources Manager LMS passes to EP_QPI (via protocol boundary 1) one product set identification message and the private correlator that it received with the request.

This process uses the private correlator passed to it with the data messages to retrieve the PRID and control point name from the original request.

EP_QPI builds subvectors from the product set identification data received from the Physical Resources Manager LMS. See "Formats Supported" on page 10-66 for the subvectors that must be built.

Using the PRID that it has just retrieved, this process now constructs a complete NMVT enveloping the Reply Product Set ID (X'0090') major vector. The sequence field in byte 7 of NMVT RU must be set to one of the following values based on information received from the Physical Resources Manager LMS with the product set identication:

- B'00', if this is the only Reply Product Set ID (X'0090') major vector for this PRID
- B'01', if this is the last Reply Product Set ID (X'0090') major vector for this PRID
- B'10', if this is the first Reply Product Set ID (X'0090') major vector for this PRID
- B'11', if this is the middle Reply Product Set ID (X'0090') major vector for this PRID

See "Building a Management Services Major Vector" on page 10-151 and "Building an NMVT" on page 10-151 for details.

This process starts the process described in "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set group) via protocol boundary A, to send the

NMVT. "Protocol Boundary A - Send NMVT" on page 8-18 shows the items passed when starting that process. This process places the following values in these items:

Item 1: The NMVT it has constructed

Item 2: The address of the control point that sent the request

After this process has started the "Sending NMVTs" process, it terminates.

EP_CHANGE_MGMT Function Set

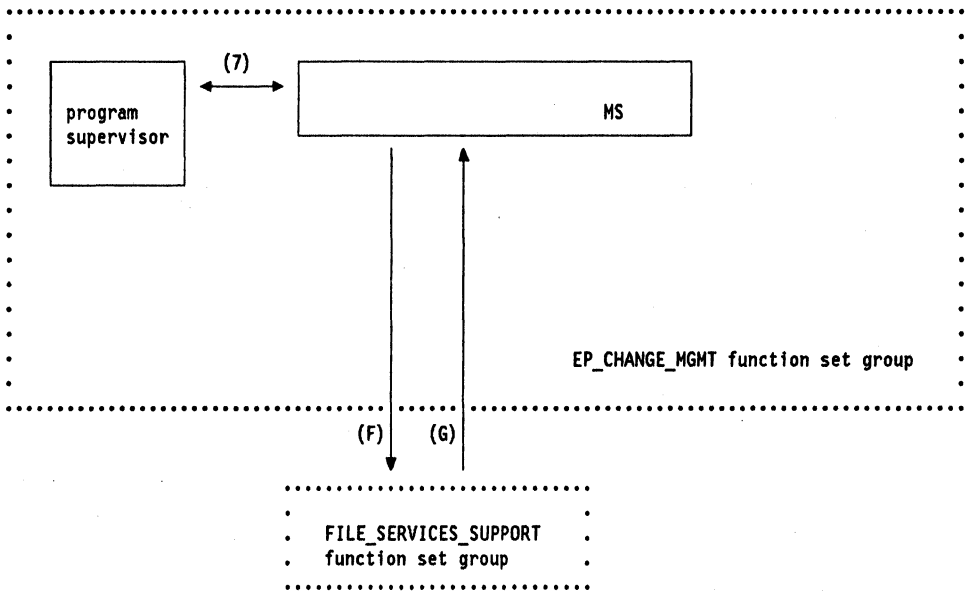


Figure 10-29. EP_CHANGE_MGMT Function Set Group

The EP_CHANGE_MGMT function set provides change management assistance to a change management focal point.

Refer to Figure 10-29 throughout the discussion of the EP_CHANGE_MGMT function set.

Protocol Boundaries with Components Outside EP_CHANGE_MGMT

- Input:
  - MS protocol boundary G (MS Bulk Data Received)

    The EP_CHANGE_MGMT function set group receives requests from the FILE_SERVICES_SUPPORT function set group. The details of this protocol boundary are described in "Protocol Boundary G - MS Bulk Data Received" on page 8-20.
- Output:
  - MS protocol boundary F (Send MS Bulk Data)

    The EP_CHANGE_MGMT function set group requests the FILE_SERVICES_SUPPORT function set group to send MS data on a SNA/DS conversation. The details of this protocol boundary are described in "Protocol Boundary F - Send MS Bulk Data" on page 8-20.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

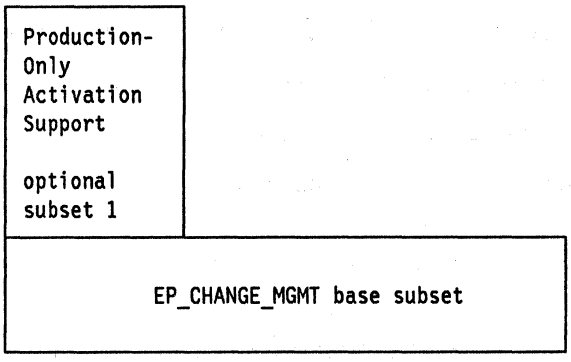


Figure 10-30. Base and Optional Subsets of EP_CHANGE_MGMT Function Set

EP_CHANGE_MGMT Base Subset

Functions Provided

The EP_CHANGE_MGMT base subset provides the capability to respond to requests from a change management focal point.

Formats Supported

The EP_CHANGE_MGMT base subset supports the receiving of a CP-MSU containing either a Request Change Control (X'8050') or a Request Activation (X'8066') major vector.

The base entry point honors only the TRIAL_AND_PRODUCTION request, as indicated in the Change Management Activation Use (X'20') subfield of the Activate (X'81') subvector of the Request Activation (X'8066') major vector, from a focal point (but either type of request from a local operator).

The following subvectors are included by this function set group in a CP-MSU containing a Change Control (X'0050') major vector as the result of processing a change control request, and passed to the FILE_SERVICES_SUPPORT function set group:

- A Reporting Installation (X'82') subvector
  - When reporting a successful or unsuccessful attempt to install a change
- A Reporting Removal (X'84') subvector

  - When reporting a successful or unsuccessful attempt to remove a change

- A Reporting Acceptance (X'86') subvector
  - When reporting a successful or unsuccessful attempt to accept a change
- One Reported Change Name (X'88') subvector for each change file referred to in the request major vector
  - To identify a change file referenced by the report
- One Detailed Data (X'98') subvector for each product-unique error
  - To report detailed product-specific data relevant to the change report
- A Reporting Secondary Installation (X'8A') subvector and one or more Secondary Change Name (X'8C') subvectors
  - To report changes not referred to in the request or its corequisite list that were installed as part of the operation being reported
- A Reporting Back-Level Status (X'8E') subvector and one or more Back-Level Change Name (X'90') subvectors
  - To report changes not referred to in the request or its corequisite list that were put into back-level state as part of the operation being reported
- A Reporting Deletion (X'92') subvector and one or more Deleted Change Name (X'94') subvectors
  - To report changes not referred to in the request or its corequisite list that were deleted as part of the operation being reported

In addition, an SNA/FS Action Summary (X'1548') GDS variable is included whenever a change control request required SNA/FS action.

The following subvector is included by this function set group in a CP-MSU containing a Reply Activation Acceptance (X'0066') major vector as the result of processing an activation request, and passed to the FILE_SERVICES_SUPPORT function set group:

- An Activation Acceptance (X'82') subvector
  - To indicate that the activation will be attempted.

For a CP-MSU containing either a Change Control (X'0050') or a Reply Activation Acceptance (X'0066') major vectors, an SNA Condition Report (X'1532') GDS variable is included to report exception conditions if they exist.

Implementation Requirements

The requirements for implementing the EP_CHANGE_MGMT base subset are described by a model consisting of a subset of the program supervisor and management services.

A Subset of the Program Supervisor:

- Interacts with MS via protocol boundary 7 as follows:
  - Processes requests from MS to control changes
    - Installs, removes, and accepts change files on request from MS and reports the result to MS
  - Processes requests from MS to reactivate the entry point.

A Subset of Management Services:

- Provides the following processes:
  - Receiving change management requests
    - Upon request by the FILE_SERVICES_SUPPORT function set group (via protocol boundary G), does the following:
      - Parses the requests for validity and does the following if an error is detected:
        - Builds a negative reply and passes it to FILE_SERVICES_SUPPORT for forwarding to the requesting LU
        - Builds and sends an Alert for exception conditions that require it
      - If the request parsed correctly, passes the request to the program supervisor
  - Sending change management data
    - Builds the CP-MSUs described in "Formats Supported" in this section and passes them along with other parameters to the FILE_SERVICES_SUPPORT function set group via protocol boundary F

Receiving Change Management Requests:

This process is started by the process described in "Receiving Requests and Data" on page 9-7 (FILE_SERVICES_SUPPORT function set), when the latter has identified the major vector key X'8050' (Request Change Control) or X'8066' (Request Activation) in a request CP-MSU.

The input to this process (the CP-MSU contents) was determined at the focal point, at the time the verb was issued, as shown in Table 10-7 on page 10-75.

Table 10-7. Choosing Major Vector/Subvector and Server Parameters from the Verb		
Verb issued by network operator	**Major Vector/ Subvector**	**FS server instructions (E=encoder D=decoder S=source T=target)**
SEND-AND-INSTALL (DESTRUCTION ALLOWED)	X'8050' / '81'	S: (FETCH, ABEND, ONLY_ IF_ EXCEPTIONS) T: (CREATE/LOAD_ OR_ REPLACE, EXECUTING, ABEND, DETAILED)
SEND-AND-INSTALL (DESTRUCTION NO)	X'8050' / '81'	S: (FETCH, ABEND, ONLY_ IF_ EXCEPTIONS) T: (CREATE/LOAD, EXECUTING, BACKOUT, DETAILED)
INSTALL	X'8050' / '81'	E: (ENCODE_ ONLY, ABEND, ONLY_ IF_ EXCEPTIONS) D: (DECODE_ ONLY, ABEND, DETAILED)
REMOVE	X'8050' / '83'	E: (ENCODE_ ONLY, ABEND, ONLY_ IF_ EXCEPTIONS) D: (DECODE_ ONLY, ABEND, DETAILED)
ACCEPT	X'8050' / '85'	E: (ENCODE_ ONLY, ABEND, ONLY_ IF_ EXCEPTIONS) D: (DECODE_ ONLY, ABEND, DETAILED)
ACTIVATE	X'8066' / '81'	SNA/FS ACTION IS NOT REQUESTED AND THERE IS NO SERVER OBJECT. SNA/DS SERVICE PARMS ARE: PRIORITY=FAST, PROTECTION=NO, CAPACITY≤4K, SECURITY=NO

Notes:

1. For all change management commands with SNA/FS action requested, TARGET_AGENT_REPORTING_ACTION = DETAILED is implied.
2. For Install, Remove, and Accept, the focal point identifies the file using a Stored_Name.
3. For Send-and-Install, the focal point identifies the file using a To-Be-Stored_Name.

The first task of this process is to determine if an SNA/FS error occurred. If so, then it builds a CP-MSU containing an SNA/FS Action Summary (X'1548') GDS variable, and terminates, returning control to the process that started it. The next task of this process is to parse the major vector. If there is a syntactic error in the major vector, or in one of the subvectors it contains, or if MS in the node fails to support some function presupposed by the request, or does not recognize the file name specified in the request, then this process builds and sends the appropriate negative reply in an SNA Condition Report (X'1532') GDS variable. The Format Exception (X'100B xxxx') report codes (as documented in *SNA Formats, GA27-3136*) are used. Report codes specific to change control are detailed in Table 10-40 on page 10-157.

If the change file named in the server object (the object of the command) is also named in the corequisite list, or if multiple instances of the same change file name occur in the list, this is ignored and processing proceeds.

A finite state machine is provided to describe how this process checks for state errors. Refer to Figure 10-31 on page 10-76. It may be invoked by both a product's SNA/FS application intervention exit to prevent unwanted storage operations, and by the the program supervisor in handling a change control request. For example, if a SEND_AND_INSTALL uses partial matching, the intervention exit will use the FSM to determine whether to replace a to-be-deleted change, then

the program supervisor will use it later to determine the state transition for the change to be installed.

STATE TRANSITION TABLE

State Names: Sent		Installed on Trial	Installed in Production Removably	Installed in Production Non-Removably	Back-Level	Reset (not present)
State numbers: Inputs:	1	2	3	4	5	6
Create/Load	► (089E0001)	► (089E0001)	► (089E0001)	► (089E0001)	► (089E0001)	1
Replace	6	► (08380001)	► (08380001)	- (b)	► (08380011)	/
Delete	6	► (08380001)	► (08380001)	6 (a)	► (08380011)	► (089A0001)
Fetch	–	–	–	–	–	► (089A0001)
Install on Trial	2	► (08380003)	► (08380004)	► (08380005)	► (08380002)	► (089A0001)
Install in Production Removably	3	3	► (08380004)	► (08380005)	► (08380002)	► (089A0001)
Install in Production Non-Removably	4	4	► (08380004)	► (08380005)	► (08380002)	► (089A0001)
Remove	► (08380007)	6	6	► (08380006)	► (08380002)	► (089A0001)
Accept	► (08380007)	► (0838000E)	4	► (08380006)	► (08380002)	► (089A0001)

"/" = indicates a should-not-occur condition
"►" = indicates exception condition (followed by exception code)
"n" = indicates state transition (output codes may be listed; see table below)

OUTPUT CODES

Output Code	Description
a	If not critical (else X'0838000F')
b	If agent object does not contain an "Install in production nonremovably" command: X'0838000F' if change is critical, X'08380013' if not If change is required to maintain removability: X'08380011'

"►" = exception condition
"–" = no state transition

Figure 10-31. EP_CHANGE_MGMT Processing State Transitions and Output Codes

Report codes specific to activation are detailed in Table 10-41 on page 10-158.

If Activate specifies use of changes installed both on-trial and in-production, then the program supervisor uses on-trial versions of components instead of in-production versions, if they exist. Therefore, it searches for a component (e.g. a microcode module) in the following order: trial version; if not available, production version; if not available, unaltered version.

If the type of activation is orderly and sessions are active, the program supervisor rejects the request. If the type is forced, activation proceeds regardless.

If no condition requiring a negative reply is found, then this process starts the program supervisor. The action taken by the program supervisor depends on the request made. The actions are illustrated in Figure 10-32, Figure 10-33 on page 10-78, and Figure 10-34 on page 10-78.

Pre-Test	IF a pre-test is required (or desired, and the program supervisor is able to perform it), THEN it is attempted. IF not successful, THEN go to "Reply".
Installation	The change is installed (along with corequisite changes, if any). IF successful, THEN go to "Post-Test".
Automatic Removal	IF automatic removal is required (or desired, and the program supervisor is able to perform it), THEN it is attempted. Go to "Reply".
Post-Test	IF a post-test is required (or desired, and this process is able to perform it), THEN it is attempted. IF not successful, THEN go to "Automatic Removal".
Automatic Acceptance	IF an automatic acceptance is prohibited THEN go to "Reply" Acceptance is attempted.
Reply	Start the process to build reply major vector.

Figure 10-32. Entry Point Processing for Install

Removal	The program supervisor attempts to remove the change. IF not successful, THEN go to "Reply".
Post-Test	IF a post-test is required (or desired, and the program supervisor is able to perform it), THEN it is attempted. The removed change file is deleted by the program supervisor.
Reply	Build reply major vector.

Figure 10-33. Entry Point Processing for Remove

Acceptance	The program supervisor attempts to accept the change. The accepted change file is deleted by the program supervisor.
Reply	Build reply major vector.

Figure 10-34. Entry Point Processing for Accept

After this process has started the program supervisor, it terminates, returning control to the process that started it.

Sending Change Management Data:

This process is started by the program supervisor to transfer change management data to MS. When this process is started it is passed this data. This process first constructs a Reporting Installation (X'82'), Reporting Removal (X'84'), Reporting Acceptance (X'86'), or Reporting Activation (X'82') subvector. It then builds other subvectors, described as follows.

A Reported Change Name (X'88') subvector is included for each change data object referenced by the request.

One or more Detailed Data (X'98') subvectors are optionally included. Within each such subvector, the only subfield handled by the base focal point is the Detailed Data (X'82') subfield. The other subfield, Qualified Message Data (X'01'), is not supported. Within each X'82' subfield, change management does not support product set ID indexing (byte 2 = X'00'), only supports the Data ID of STATUS CODE (byte 3 = X'13'), and a Data Encoding of EBCDIC character set 00640−00500 plus (byte 4 = X'11').

A Reporting Secondary Installation (X'8A') subvector, and one or more Secondary Change Name (X'8C') subvectors, is included if a change not referred to in the request (or in its corequisite list) was installed as part of the operation being reported.

A Reporting Back-Level Status (X'8E') subvector, and one or more Back-Level Change Name (X'90') subvectors, is included if a change not referred to in the request (or in its corequisite list) was put into back-level state as part of the operation being reported.

A Reporting Deletion (X'92') subvector, and one or more Deleted Change Name (X'94') subvectors, is included if a change not referred to in the request (or in its corequisite list) was deleted as part of the operation being reported. Such deletions are change management, not SNA/FS, deletions. SNA/FS deletions are reported in the server object using DELETED_TOKEN_STRING.

The following table shows when secondary installation, back-level status, and deletions are reported.

Table 10-8. Reporting Subvectors Used Based on State of Primary Change Files

Resultant state of primary change files	**Subvectors included in the report**
Installed on trial	(X'82', X'88's)
Installed in production removably	(X'82', X'88's), (X'8E', X'90's)
Installed in production non-removably	(X'82', X'88's), (X'92', X'94's)
Removed	(X'84', X'88's), (X'8A', X'8C's) if transiting from installed in production removably, not installed on trial
Accepted	(X'86', X'88's), (X'92', X'94's)
Deleted (explicitly)	SNA/FS deleted token string is used for reporting

Notes:

1. X'92' and X'94' subvectors are used to report change management actions. If an SNA/FS action deleted a change file (for example, in partial matching where the old version is deleted), then the subvectors are not used since such deletion is reported in the server object.

This process then constructs the common subvectors, builds a complete CP-MSU, logs it for later reference (e.g. by a local operator during problem diagnosis), and then starts the process described in "Sending Requests and Data" on page 9-6 (FILE_SERVICES_SUPPORT function set group via protocol boundary F) to send the report to the requesting LU. SNA/FS parameters are also returned, if they were provided on the request, to indicate that a file was transferred in the same MU as the CP-MSU. This will only occur if an Install (X'81') subvector was in a Request Change Control (X'8050') major vector. These parameters are shown in Table 10-9 on page 10-80.

Table 10-9. Choosing SNA/FS Server Parameters in the Report Direction		
Condition	**Major Vector/ Subvector**	**SNA/FS server instructions (E=encoder D=decoder S=source T=target)**
FILE TRANSFERRED WITH REQUEST	X'8050' / '81'	E: (ENCODE_ ONLY, ABEND, ONLY_ IF_ EXCEPTIONS) D: (DECODE_ ONLY, ABEND, DETAILED)
FILE NOT TRANSFERRED WITH REQUEST	X'8050' / '81'	NONE SPECIFIED (NO SERVER OBJECT IN REPORT MU)
FILE IS NEVER TRANS-FERRED WITH REQUEST	X'8050' / '83'	NONE SPECIFIED (NO SERVER OBJECT IN REPORT MU)
FILE IS NEVER TRANS-FERRED WITH REQUEST	X'8050' / '85'	NONE SPECIFIED (NO SERVER OBJECT IN REPORT MU)
FILE IS NEVER TRANS-FERRED WITH REQUEST	X'8066' / '81'	NONE SPECIFIED (NO SERVER OBJECT IN REPORT MU). SNA/DS SERVICE PARMS ARE: PRIORITY=FAST, PROTECTION=NO, CAPACITY≤4K, SECURITY=NO.

This process then terminates.

EP_CHANGE_MGMT Optional Subset 1 (Production-Only Activation Support)

Functions Provided

EP_CHANGE_MGMT Optional Subset 1 responds to requests from the focal point for activation of only those versions of components marked in-production.

Formats Supported

This optional subset supports:

- Receiving the Production Only (X'20') value of the Change Management Activation Use (X'20') subfield of the Activate (X'81') subvector of the Request Activation (X'8066') major vector

Implementation Requirements

The details for implementing major vectors containing the subfields used by this subset are described in the base subset.

Example Flows

Non-Destructive SEND_AND_INSTALL

Consider the three nodes in a SNA/DS network depicted below. Assume that all three nodes happen to be in the same network (NET1).

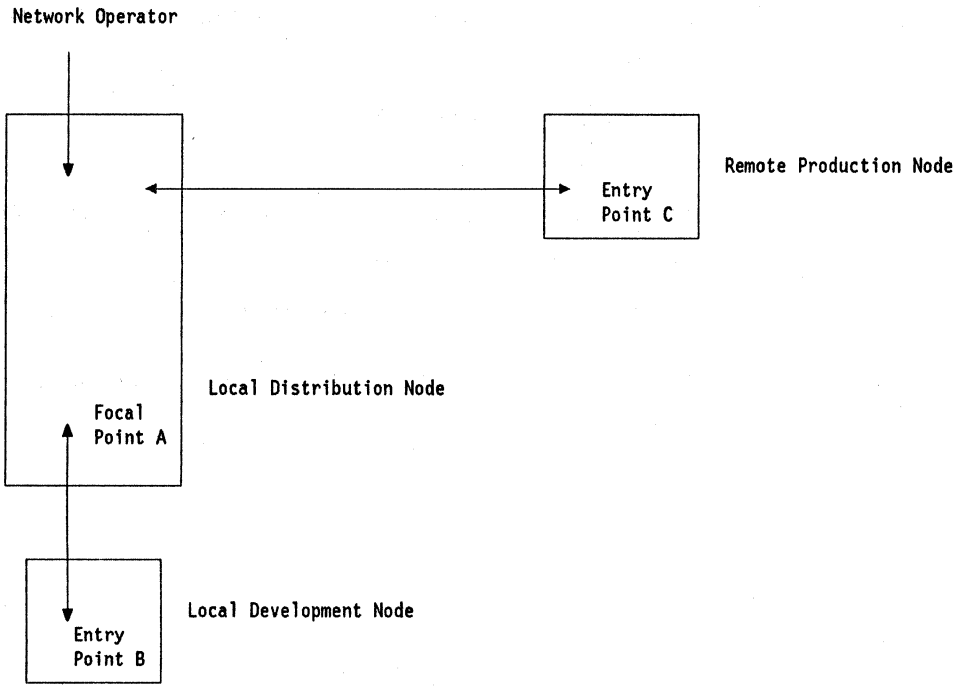


Figure 10-35. Sample configuration

Given the configuration in Figure 10-35, in the example that follows, a microcode customizing data change file with SNA/FS global name MCUST.9135.NA.NET1.C.SYSTEM.V2 has been prepared at a development node (entry point B) and sent, in an unsolicited manner, to a distribution node (focal point A).

A network operator at A wishes to send and install the change file at a remote production node (entry point C). He chooses the following installation options:

- Trial installation
- Removability required
- Pre-test required
- Automatic Removal if test failure or installation failure
- Post-test not to be performed
- Automatic Acceptance prohibited

A discussion of these Install options can be found in "Change Control" on page 6-5.

Network operator at A issues request:

In the architected model, the network operator then assigns a sequence number and issues the Send_and_Install Change Management request protocol boundary verb which is defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-10). However, a typical implementation will assign the sequence number automatically before issuing the verb.

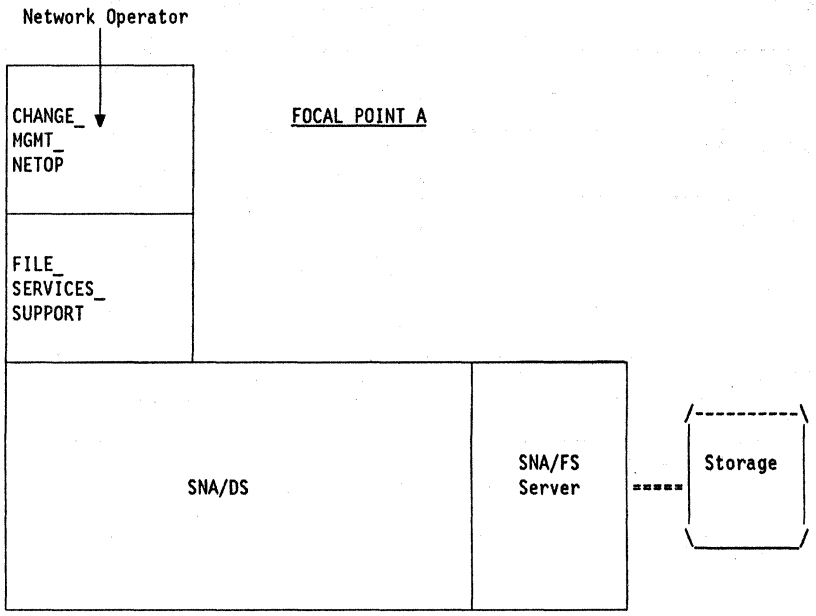


Figure 10-36. Non-Destructive Send_and_Install (1 of 6). Network operator at A issues Send_and_Install verb to CHANGE_MGMT_NETOP.

The parameters supplied on the Send_and_Install verb by the network operator are:

Table 10-10. Send_and_Install Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'203'	CORRELATION VALUE FOR THIS UNIT OF WORK
TO_BE_FETCHED_NAME	MCUST, 9135, NA, NET1, C, SYSTEM, V2	SNA/FS GLOBAL NAME OF FILE TO BE FETCHED
TARGET_LIST TARGET_LOCATION	 NET1.C	LOCATION(S) TO SEND CHANGE FILE
DESTRUCTION	NO	PROHIBIT SNA/FS FROM OVER-WRITING ANY FILES
REMOVABILITY	YES	INSTALL FILE SO IT CAN BE PROC-ESSED BY SUBSEQUENT REMOVE PROTOCOL BOUNDARY VERB
AUTOMATIC_REMOVAL	YES	REMOVE CHANGE FILE IF INSTALLA-TION OR TEST FAILS
PRE_TEST	YES	PERFORM PRE-TEST
POST_TEST	NO	DO NOT PERFORM POST-TEST
AUTOMATIC_ACCEPTANCE	NO	DO NOT AUTOMATICALLY ACCEPT CHANGE FILE IF INSTALLATION AND TEST SUCCEED
ACTIVATION_USE	TRIAL	COMPONENTS ALTERED BY INSTAL-LATION WILL BE USED IN LATER TRIAL ACTIVATION

Request message unit is built at A and sent to C:

CHANGE_MGMT_NETOP at A builds the request CP-MSU, identifies the target destination as NET1.C, chooses server parameters and assigns an agent unit-of-work correlator. It then passes this data to FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-11).

As a result, the SNA/FS server fetches the change file and encodes the server object. SNA/DS then builds and sends a message unit (MU) to remote node C.

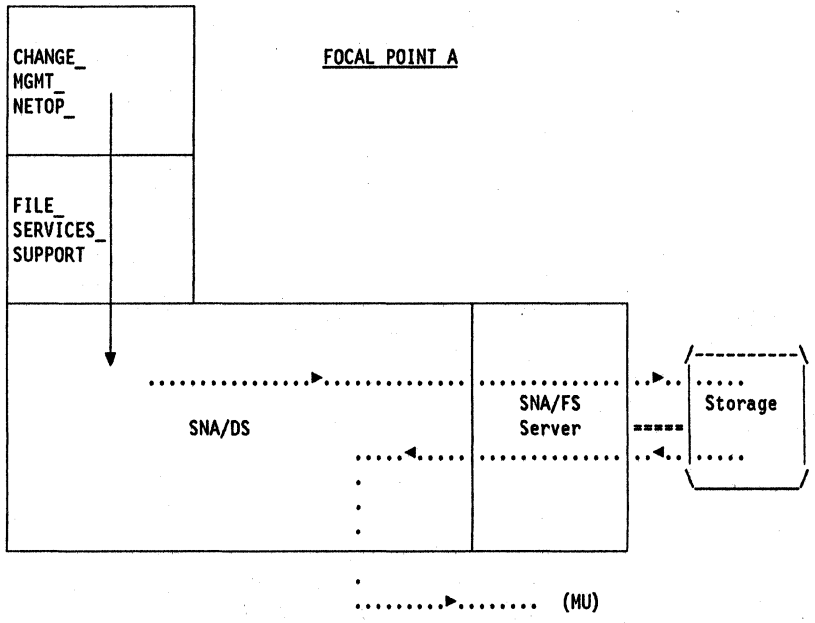


Figure 10-37. Non-Destructive Send_and_Install (2 of 6). CHANGE_MGMT_NETOP at A invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-11. Send_Distribution Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	1234	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.C	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
SOURCE_INSTRUCTION	FETCH ABEND ONLY_IF_EXCEPTIONS	
TARGET_INSTRUCTION	CREATE/LOAD EXECUTING BACKOUT DETAILED	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	TO_BE_FETCHED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V2)	

Request message unit is received at C:

SNA/DS at C receives the message unit and the SNA/FS server decodes the server object and stores the change file. Then SNA/DS invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-12).

FILE_SERVICES_SUPPORT then invokes EP_CHANGE_MGMT, passing it the CP-MSU, source destination (NET1.A), server parameters and agent unit-of-work correlator.

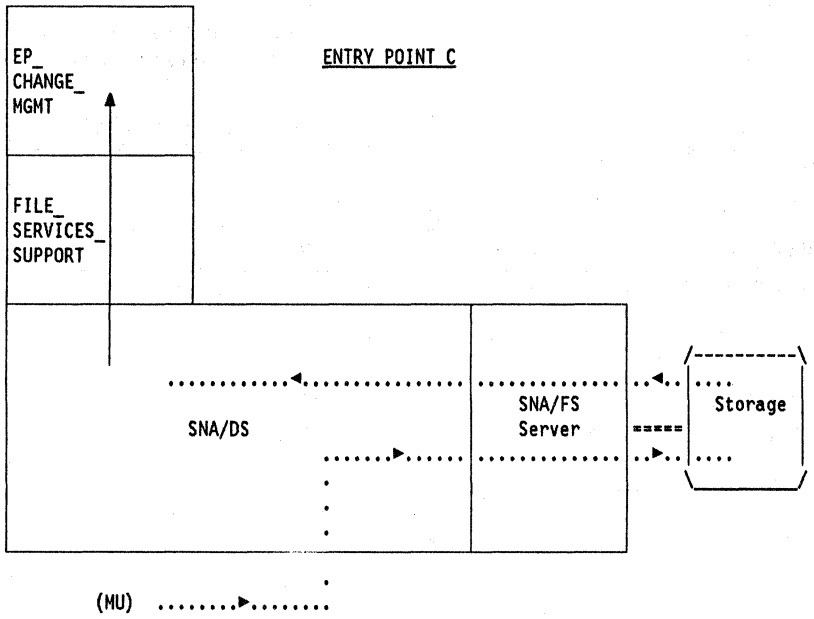


Figure 10-38. Non-Destructive Send_and_Install (3 of 6). FILE_SERVICES_SUPPORT at C issues Receive_Distribution and is returned parameters. It then invokes EP_CHANGE_MGMT.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-12. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	1234	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
SERVER_SUMMARY_REPORT	ALL_SUCCESSFUL	
TARGET_INSTRUCTION	CREATE/LOAD EXECUTING BACKOUT DETAILED	
D_O_TYPE	X'10200000'	
D_O_LOCAL_NAME	LOCAL.NAME	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V2)	

Response message unit is built at C and sent to A:

EP_CHANGE_MGMT at C parses the CP-MSU, extracting the Request Change Control major vector. Based on the Install subvector encodings, a pre-test is performed successfully on the change file and it is installed on trial removably. To report installation, it builds a reply CP-MSU, identifies the target destination as NET1.A, chooses server parameters and specifies the agent unit-of-work correlator. Finally, it invokes FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-13).

As a result, the SNA/FS server encodes a new server object and SNA/DS builds and sends a message unit back to node A.

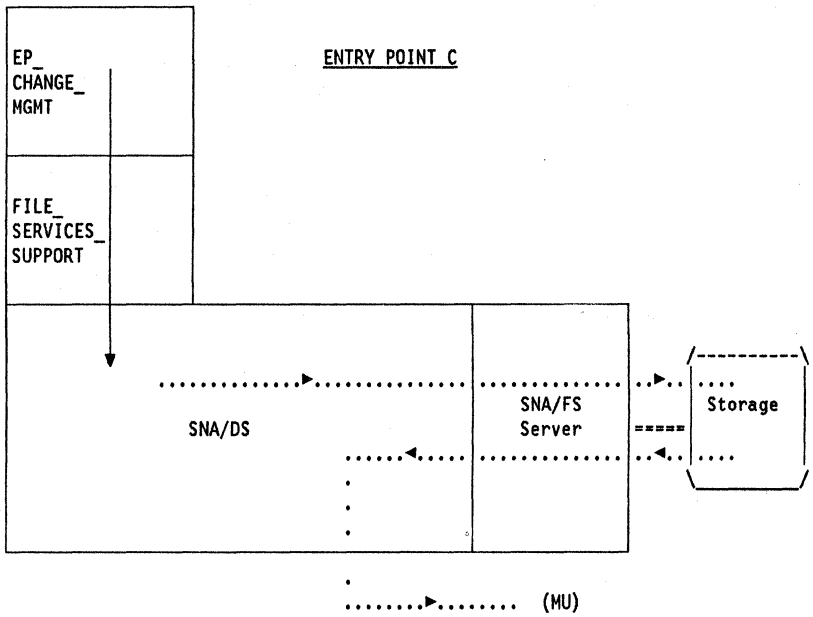


Figure 10-39. Non-Destructive Send_and_Install (4 of 6). EP_CHANGE_MGMT at C invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-13. Send_Distribution Supplied Parameters		
Parameter Name	**Parameter Value**	**Parameter Meaning**
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	5678	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.A	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: CHANGE CONTROL MAJOR VECTOR CONSISTING OF: DATE/TIME SUBVECTOR REPORTING INSTALLATION SUBVECTOR REPORTED CHANGE NAME SUBVECTOR FS ACTION SUMMARY GDS	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
ENCODER_INSTRUCTION	ENCODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V2)	

Response message unit is received at A:

SNA/DS at A receives the message unit and the SNA/FS server decodes the server object. Then SNA/DS invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-14).

FILE_SERVICES_SUPPORT then invokes CHANGE_MGMT_NETOP, passing it the CP-MSU, source destination (NET1.C), server parameters and agent unit-of-work correlator.

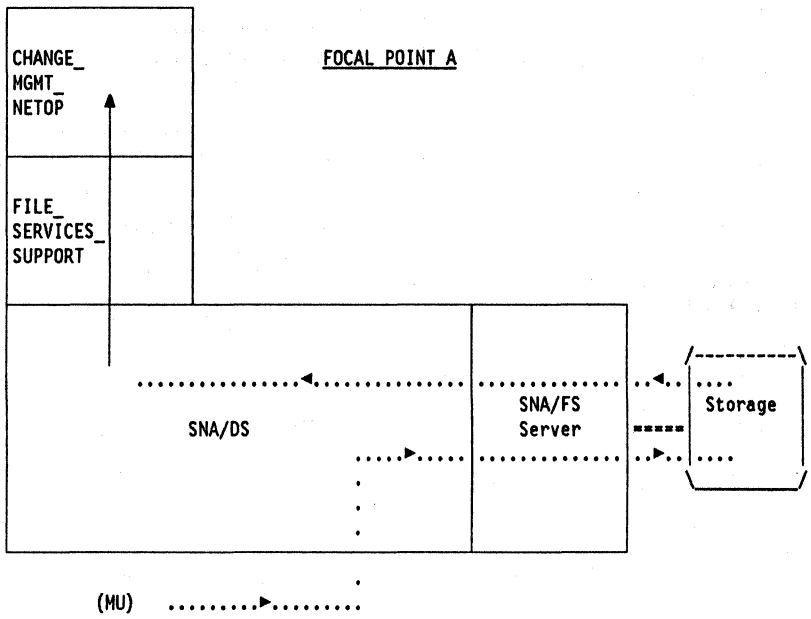


Figure 10-40. Non-Destructive Send_and_Install (5 of 6). FILE_SERVICES_SUPPORT at A issues Receive_Distribution and is returned parameters. It then invokes CHANGE_MGMT_NETOP.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-14. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	5678	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: CHANGE CONTROL MAJOR VECTOR CONSISTING OF: DATE/TIME SUBVECTOR REPORTING INSTALLATION SUBVECTOR REPORTED CHANGE NAME SUBVECTOR FS ACTION SUMMARY GDS	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V2)	

Network operator at A is issued response:

After parsing the reply CP-MSU and extracting data, CHANGE_MGMT_NETOP at A builds and issues, to the network operator, the Reporting_Installation protocol boundary verb for Change Management defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-15). Note that the Correlator value is the same as that originally provided by the network operator on the Send_and_Install verb.

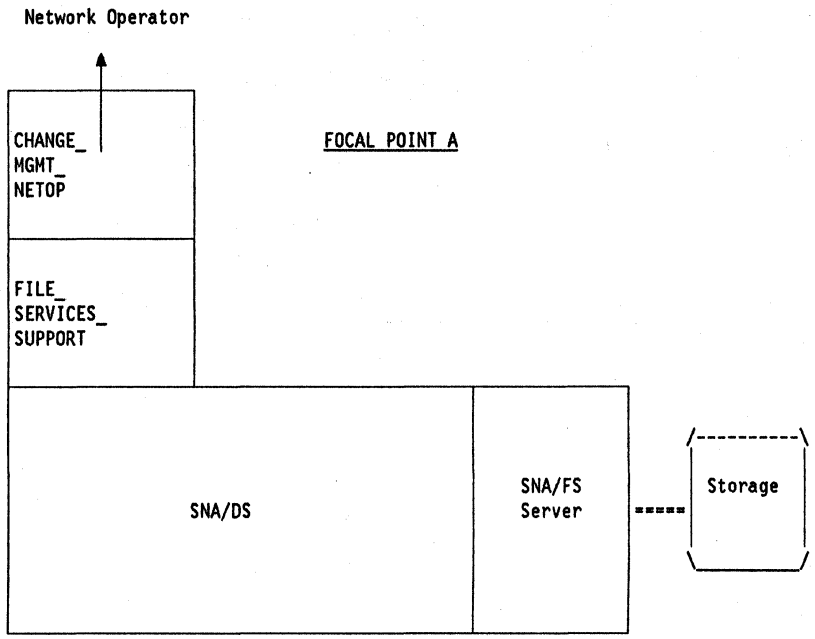


Figure 10-41. Non-Destructive Send_and_Install (6 of 6). CHANGE_MGMT_NETOP at A issues Reporting_Installation verb to network operator.

The parameters returned on the Reporting_Installation verb to the network operator are:

Table 10-15. Reporting_Installation Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'203'	CORRELATES REPORT TO INITIAL REQUEST
TIME_STAMP	DATE/TIME OF INSTALLATION	FROM DATE/TIME SUBVECTOR IN RETURNED CP-MSU
TARGET_LOCATION	NET1.C	LOCATION WHERE FILE STORED
STORED_NAME	MCUST, 9135, NA, NET1, C, SYSTEM, V2	SNA/FS GLOBAL NAME OF STORED FILE
FS_ACTION_SUMMARY	ALL_SUCCESSFUL NO_BACKOUT_ATTEMPTED ABEND_NOT_APPLICABLE CONTINUE_NOT_ATTEMPTED	AS DEFINED BY SNA/FS
INSTALLATION_RESULTS INSTALLATION_STATUS WHEN_EFFECTIVE REPORTED_CHANGE_NAME_LIST CHANGE_FILE_NAME	 SUCCESSFUL ACTIVATION_REQUIRED MCUST, 9135, NA, NET1, C, SYSTEM, V2	 INSTALLATION SUCCESSFUL UPON ACTIVATION CHANGE FILE(S) INSTALLED
PRE_TEST_STATUS	SUCCESSFUL	TEST WAS SUCCESSFUL
POST_TEST_STATUS	NOT_ATTEMPTED	TEST NOT ATTEMPTED
AUTOMATIC_REMOVAL_RESULTS AUTOMATIC_REMOVAL_STATUS WHEN_EFFECTIVE	 NOT_ATTEMPTED NOT_APPLICABLE	 REMOVAL NOT ATTEMPTED EFFECTIVE TIME NOT APPLICABLE
AUTOMATIC_ACCEPTANCE_STATUS	NOT_ATTEMPTED	ACCEPTANCE NOT ATTEMPTED
REMOVABILITY_STATUS	INSTALLED_REMOVABLY	CHANGE FILE CAN BE REMOVED
ACTIVATION_USE_STATUS	TRIAL	CHANGE FILE INSTALLED ON TRIAL

Failing Non-Destructive SEND_AND_INSTALL

Consider the three nodes in a SNA/DS network depicted below. Assume that all three nodes happen to be in the same network (NET1).

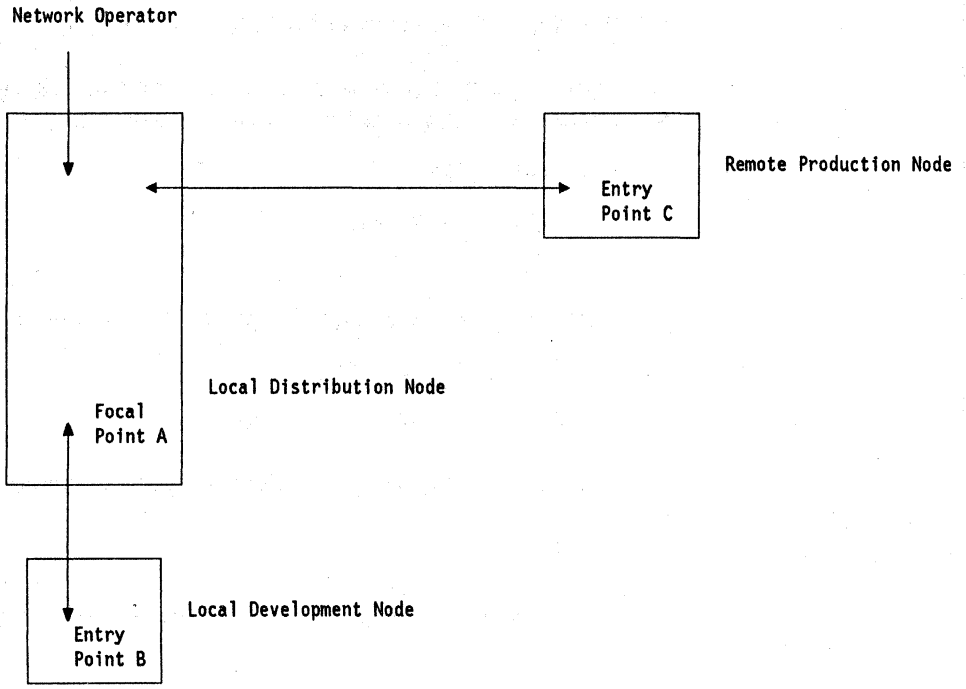


Figure 10-42. Sample Configuration

Given the configuration in Figure 10-42, in the example that follows, a microcode customizing data change file with SNA/FS global name MCUST.9135.NA.NET1.C.SYSTEM.V3 has been prepared at a development node (entry point B) and sent, in an unsolicited manner, to a distribution node (focal point A). This change file is a new version (V3) of the customizing data change file (V2) successfully installed in the previous example.

A network operator at A wishes to send and install the change file at a remote production node (entry point C). He chooses the following installation options:

- Trial installation
- Removability required
- Pre-test required
- Automatic Removal if test failure or installation failure
- Post-test not to be performed
- Automatic Acceptance prohibited

A discussion of these Install options can be found in "Change Control" on page 6-5.

In this example, however, the remote production node (entry point C) does not have enough storage space to accommodate the new change file.

Network operator at A issues request:

In the architected model, the network operator then assigns a sequence number and issues the Send_and_Install Change Management request protocol boundary verb which is defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-16). However, a typical implementation will assign the sequence number automatically before issuing the verb.

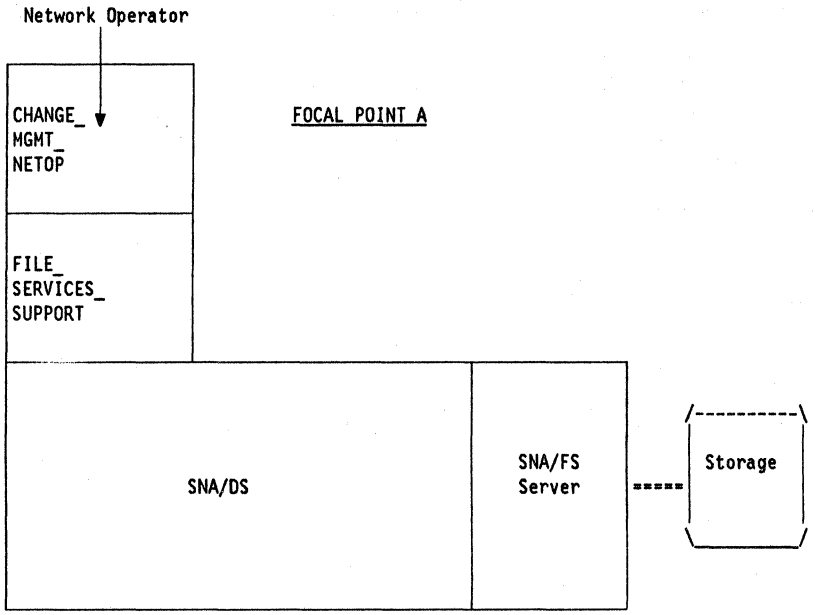


Figure 10-43. Failing Non-Destructive Send_and_Install (1 of 6). Network operator at A issues Send_and_Install verb to CHANGE_MGMT_NETOP.

The parameters supplied on the Send_and_Install verb by the network operator are:

Table 10-16. Send_and_Install Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'497'	CORRELATION VALUE FOR THIS UNIT OF WORK
TO_BE_FETCHED_NAME	MCUST, 9135, NA, NET1, C, SYSTEM, V3	SNA/FS GLOBAL NAME OF FILE TO BE FETCHED
TARGET_LIST TARGET_LOCATION	 NET1.C	LOCATION(S) TO SEND CHANGE FILE
DESTRUCTION	NO	PROHIBIT SNA/FS FROM OVER-WRITING ANY FILES
REMOVABILITY	YES	INSTALL FILE SO IT CAN BE PROC-ESSED BY SUBSEQUENT REMOVE PROTOCOL BOUNDARY VERB
AUTOMATIC_REMOVAL	YES	REMOVE CHANGE FILE IF INSTALLA-TION OR TEST FAILS
PRE_TEST	YES	PERFORM PRE-TEST
POST_TEST	NO	DO NOT PERFORM POST-TEST
AUTOMATIC_ACCEPTANCE	NO	DO NOT AUTOMATICALLY ACCEPT CHANGE FILE IF INSTALLATION AND TEST SUCCEED
ACTIVATION_USE	TRIAL	COMPONENTS ALTERED BY INSTAL-LATION WILL BE USED IN LATER TRIAL ACTIVATION

Request message unit is built at A and sent to C:

CHANGE_MGMT_NETOP at A builds the request CP-MSU, identifies the target destination as NET1.C, chooses server parameters and assigns an agent unit-of-work correlator. It then passes this data to FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-17).

As a result, the SNA/FS server fetches the change file and encodes the server object. SNA/DS then builds and sends a message unit (MU) to remote node C.

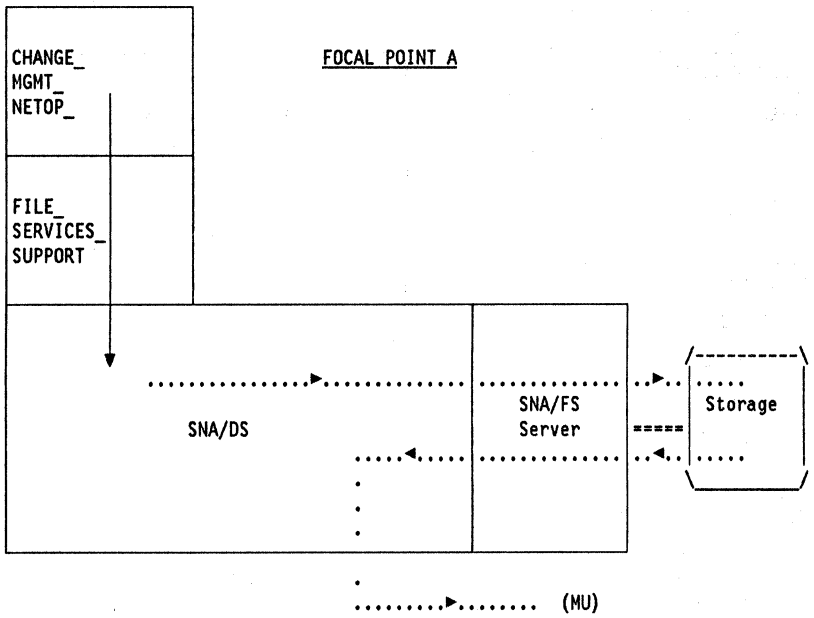


Figure 10-44. Failing Non-Destructive Send_and_Install (2 of 6). CHANGE_MGMT_NETOP at A invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-17. Send_Distribution Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	9012	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.C	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
SOURCE_INSTRUCTION	FETCH ABEND ONLY_IF_EXCEPTIONS	
TARGET_INSTRUCTION	CREATE/LOAD EXECUTING BACKOUT DETAILED	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	TO_BE_FETCHED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3)	

Request message unit is received at C:

SNA/DS at C receives the message unit and the SNA/FS server decodes the server object and in attempting to store the change file discovers there is not enough storage space to accommodate the new version. An SNA/FS data object exception report with a CREATING_ALLOCATION_EXCEPTION (X'084C0002') is returned to SNA/DS. SNA/DS then invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-18).

FILE_SERVICES_SUPPORT then invokes EP_CHANGE_MGMT, passing it the CP-MSU, source destination (NET1.A), server parameters and agent unit of work correlator.

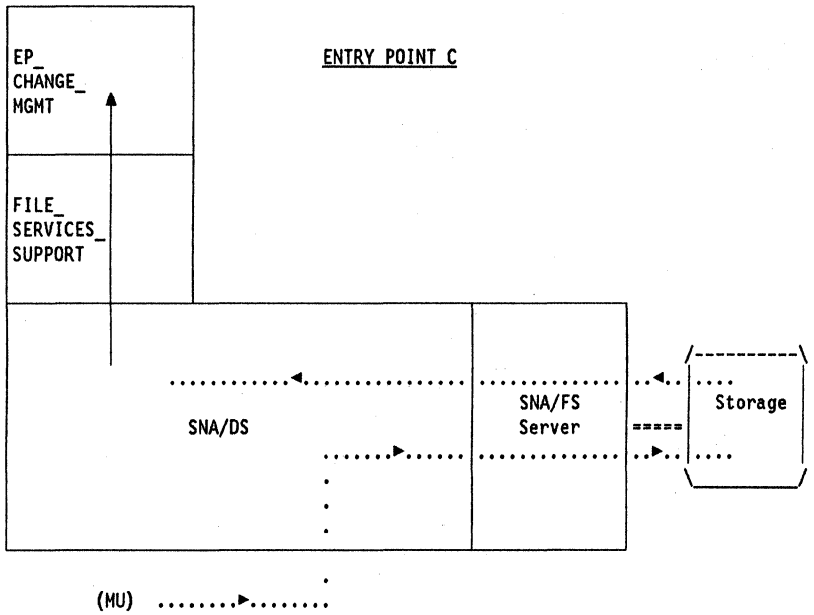


Figure 10-45. Failing Non-Destructive Send_and_Install (3 of 6). FILE_SERVICES_SUPPORT at C issues Receive_Distribution and is returned parameters. It then invokes EP_CHANGE_MGMT.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-18. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	9012	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
SERVER_SUMMARY_REPORT	NONE_SUCCESSFUL	
	ALL_BACKED_OUT	
	ABEND_NOT_APPLICABLE	
	CONTINUE_NOT_ATTEMPTED	
TARGET_INSTRUCTION	CREATE/LOAD	
	EXECUTING	
	BACKOUT	
	DETAILED	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	TO_BE_STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3)	
SNA_CONDITION_REPORT		
SNA_REPORT_CODE	X'084C0002'	

Response message unit is built at C and sent to A:

EP_CHANGE_MGMT discovers that an SNA/FS exception code is present in the server parameters. As a result, the CP-MSU is not parsed and the Install command in the Request Change Control major vector is not processed. Instead EP_CHANGE_MGMT builds a reply CP-MSU containing the FS Action Summary, identifies the target destination as NET1.A, chooses server parameters and specifies the agent unit-of-work correlator. Finally, it invokes FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-19).

As a result, the SNA/FS server encodes a new server object and SNA/DS builds and sends a message unit back to node A.

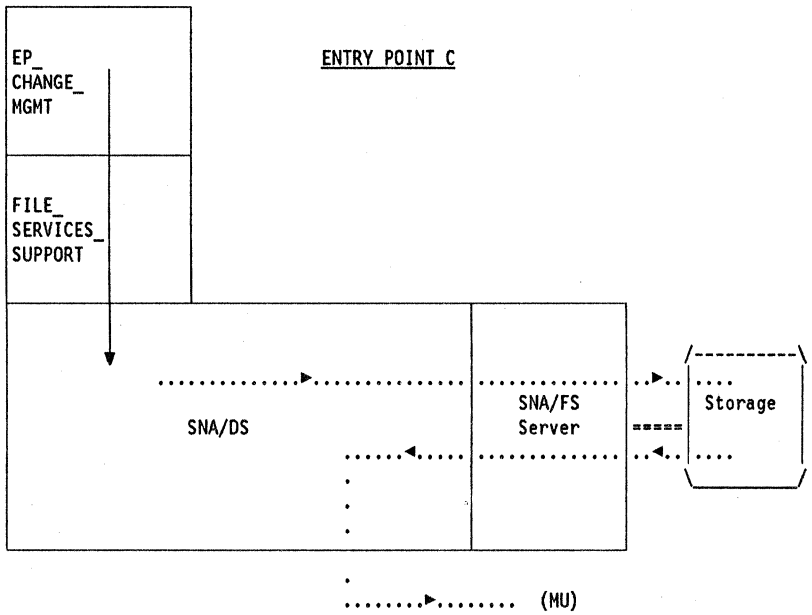


Figure 10-46. Failing Non-Destructive Send_and_Install (4 of 6). EP_CHANGE_MGMT at C invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-19. Send_Distribution Supplied Parameters		
Parameter Name	**Parameter Value**	**Parameter Meaning**
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	3456	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.A	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: FS ACTION SUMMARY GDS	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
ENCODER_INSTRUCTION	ENCODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	TO_BE_STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3)	
SNA_CONDITION_REPORT		
SNA_REPORT_CODE	X'084C0002'	
SUPPLEMENTAL_REPORT	TARGET_INSTRUCTION (CREATE/LOAD, EXECUTING, BACKOUT, DETAILED)	

Response message unit is received at A:

SNA/DS at A receives the message unit and the SNA/FS server decodes the server object. Then SNA/DS invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-20).

FILE_SERVICES_SUPPORT then invokes CHANGE_MGMT_NETOP, passing it the CP-MSU, source destination (NET1.C), distribution time, server parameters and agent unit-of-work correlator.

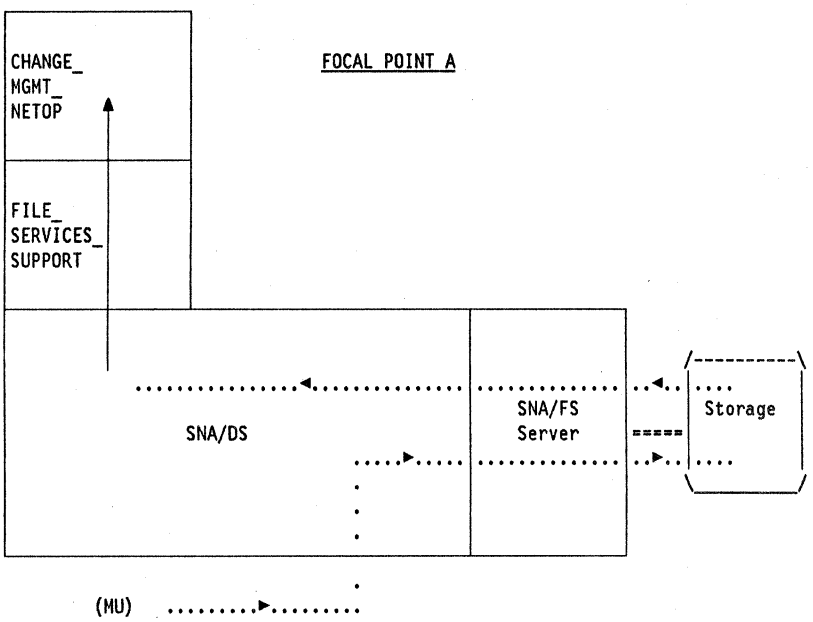


Figure 10-47. Failing Non-Destructive Send_and_Install (5 of 6). FILE_SERVICES_SUPPORT at A issues Receive_Distribution and is returned parameters. It then invokes CHANGE_MGMT_NETOP.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-20. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	3456	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF:	SEE *SNA Formats, GA27-3136*
	FS ACTION SUMMARY GDS	
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	TO_BE_STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3)	
SNA_CONDITION_REPORT		
SNA_REPORT_CODE	X'084C0002'	
SUPPLEMENTAL_REPORT	TARGET_INSTRUCTION (CREATE/LOAD, EXECUTING, BACKOUT, DETAILED)	

Network operator at A is issued response:

Based on the error identified in the FS_Action_Summary of the reply CP-MSU, CHANGE_MGMT_NETOP at A builds and issues, to the network operator, a Reply_To_Send protocol boundary verb for File Services defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-21). Note that the Correlator value is the same as that originally provided by the network operator on the Send_and_Install verb.

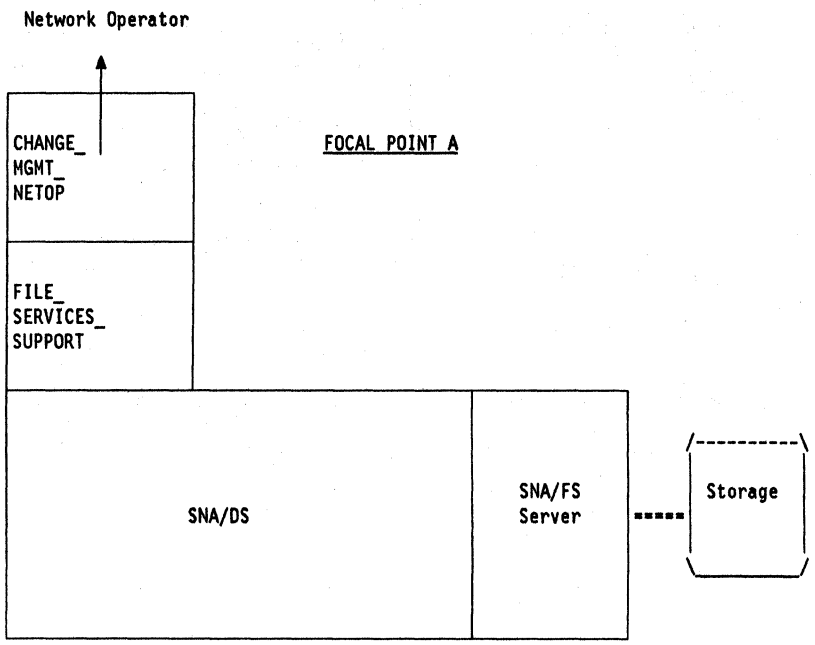


Figure 10-48. Failing Non-Destructive Send_and_Install (6 of 6). CHANGE_MGMT_NETOP at A issues Reply_to_Send verb to network operator.

The parameters returned on the Reply_To_Send verb to the network operator are:

Table 10-21. Reply_to_Send Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'497'	CORRELATES REPORT TO INITIAL REQUEST
TIME_STAMP	DATE/TIME OF INSTALLATION	FROM THE SNA/DS DISTRIBUTION_TIME PARAMETER
TARGET_LOCATION	NET1.C	TARGET OF ATTEMPTED SEND
FS_ACTION_SUMMARY	NONE_SUCCESSFUL ALL_BACKED_OUT ABEND_NOT_APPLICABLE CONTINUE_NOT_ATTEMPTED	AS DEFINED BY SNA/FS
SNA_CONDITION_REPORT		
SNA_REPORT_CODE	X'084C0002'	CREATE/LOAD ALLOCATION EXCEPTION CODE
REPORTED_ON_TOKEN_STRING	MCUST, 9135, NA, NET1, C, SYSTEM, V3	SNA/FS GLOBAL NAME OF CREATE/LOAD ERROR FILE

Destructive SEND_AND_INSTALL

Consider the three nodes in a SNA/DS network depicted below. Assume that all three nodes happen to be in the same network (NET1).

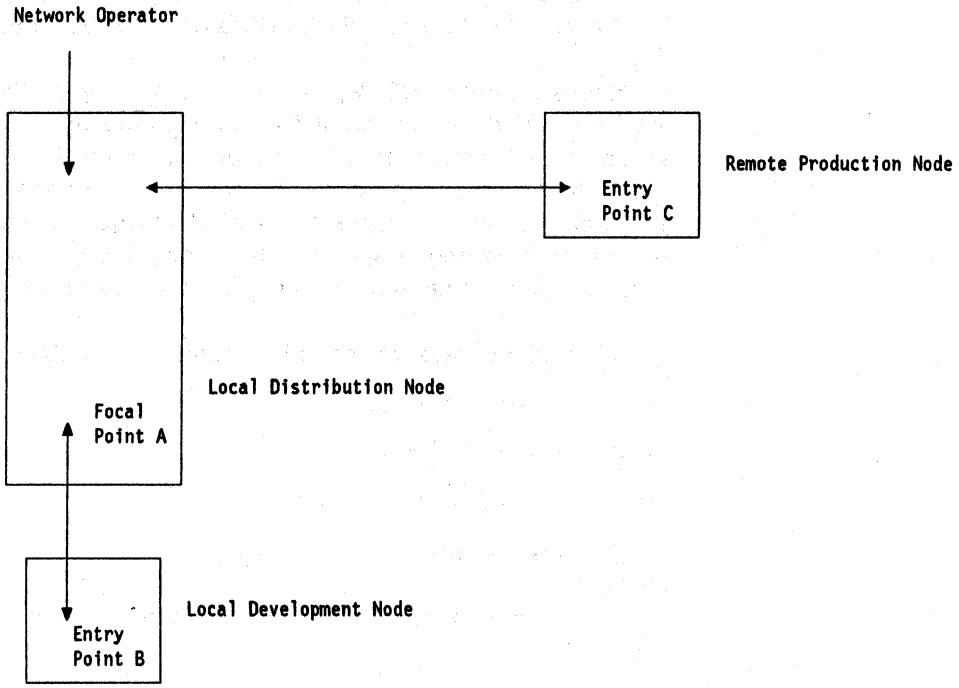


Figure 10-49. Sample Configuration

Given the configuration in Figure 10-49, in the example that follows, a microcode customizing data change file with SNA/FS global name MCUST.9135.NA.NET1.C.SYSTEM.V3 has been prepared at a development node (entry point B) and sent, in an unsolicited manner, to a distribution node (focal point A). This change file is the same version (V3) of the customizing data change file that could not be successfully stored in the previous example.

A network operator at A wishes to send and install the change file at a remote production node (entry point C), but knowing that there is insufficient storage space at the remote node he wishes to direct the SNA/FS server at C in the destruction of an existing version of the microcode customizing data change file. Specifically, he wishes to reclaim storage currently occupied by the oldest version of the same change file, that being the one having the lowest character value in token 7 (the version number) of its SNA/FS canonical identifier.

The network operator chooses the following installation options:

- Production installation
- Removability prohibited
- Pre-test required
- Post-test required

A discussion of these Install options can be found in "Change Control" on page 6-5.

Network operator at A issues request:

In the architected model, the network operator then assigns a sequence number and issues the Send_and_Install Change Management request protocol boundary verb which is defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-22). However, a typical implementation will assign the sequence number automatically before issuing the verb.

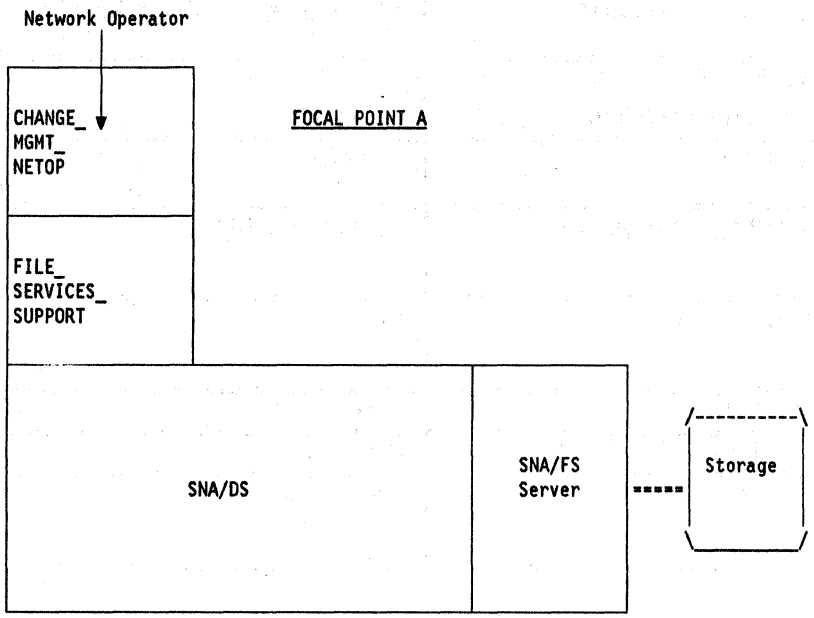


Figure 10-50. Destructive Send_and_Install (1 of 6). Network operator at A issues Send_and_Install verb to CHANGE_MGMT_NETOP.

The parameters supplied on the Send_and_Install verb by the network operator are:

Table 10-22. Send_and_Install Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'723'	CORRELATION VALUE FOR THIS UNIT OF WORK
TO_BE_FETCHED_NAME	MCUST, 9135, NA, NET1, C, SYSTEM, V3	SNA/FS GLOBAL NAME OF FILE TO BE FETCHED
TARGET_LIST TARGET_LOCATION	 NET1.C	LOCATION(S) TO SEND CHANGE FILE
DESTRUCTION	ALLOWED	ALLOW SNA/FS TO OVERWRITE FILES
DELETING_MATCH_FLAGS	(7, SELECT_LOWEST)	DELETE CHANGE FILE HAVING SAME FIRST 6 TOKENS AND LOWEST VALUE IN SEVENTH TOKEN
REMOVABILITY	NO	DO NOT ALLOW FILE TO BE PROCESSED BY SUBSEQUENT REMOVE PROTOCOL BOUNDARY VERB
PRE_TEST	DESIRED	PERFORM PRE-TEST, IF POSSIBLE
POST_TEST	DESIRED	PERFORM POST-TEST, IF POSSIBLE
ACTIVATION_USE	PRODUCTION	COMPONENTS ALTERED BY INSTALLATION MAY BE USED IN ANY TYPE OF ACTIVATION

Request message unit is built at A and sent to C:

CHANGE_MGMT_NETOP at A builds the request CP-MSU, identifies the target destination as NET1.C, chooses server parameters and assigns an agent unit-of-work correlator. It then passes this data to FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-23).

As a result, the SNA/FS server fetches the change file and encodes the server object. SNA/DS then builds and sends a message unit (MU) to remote node C.

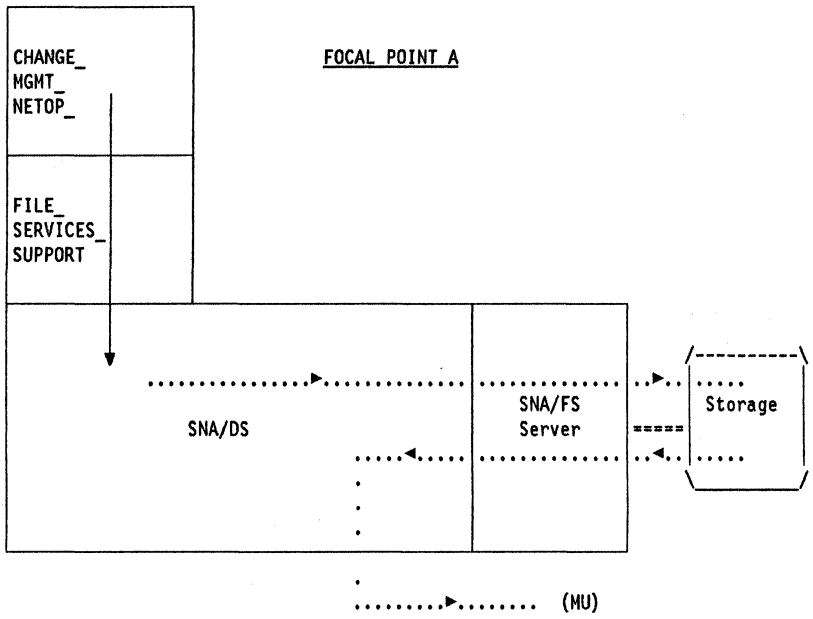


Figure 10-51. Destructive Send_and_Install (2 of 6). CHANGE_MGMT_NETOP at A invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are (Notice that the CREATE/LOAD_OR_REPLACE instruction is to be used by the SNA/FS target server, thus allowing a previous version of the change file to be overwritten):

Table 10-23. Send_Distribution Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	7890	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.C	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
SOURCE_INSTRUCTION	FETCH ABEND ONLY_IF_EXCEPTIONS	
TARGET_INSTRUCTION	CREATE/LOAD_OR_REPLACE EXECUTING ABEND DETAILED	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	TO_BE_FETCHED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3) DELETING_MATCH_FLAGS (7, SELECT_LOWEST)	

Request message unit is received at C:

SNA/DS at C receives the message unit and the SNA/FS server decodes the server object. Since the server locates two previous versions (MCUST.9135.NA.NET1.C.SYSTEM.V1 and MCUST.9135.NA.NET1.C.SYSTEM.V2) of the change file identified in the object identifier, it replaces MCUST.9135.NA.NET1.C.SYSTEM.V1 based on the criteria specified in the server parameters. Then SNA/DS invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-24).

FILE_SERVICES_SUPPORT then invokes EP_CHANGE_MGMT, passing it the CP-MSU, source destination (NET1.A), server parameters and agent unit-of-work correlator.

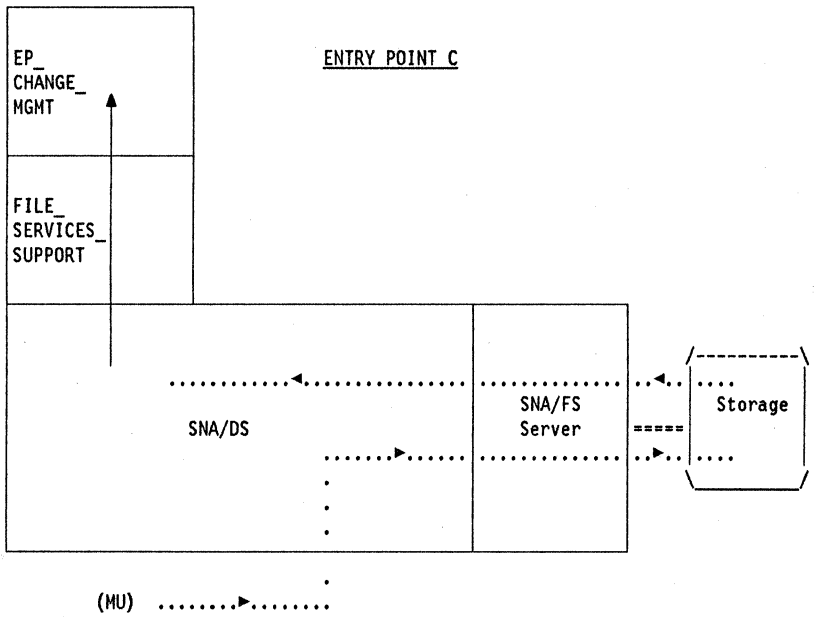


Figure 10-52. Destructive Send_and_Install (3 of 6). FILE_SERVICES_SUPPORT at C issues Receive_Distribution and is returned parameters. It then invokes EP_CHANGE_MGMT.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-24. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	7890	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
SERVER_SUMMARY_REPORT	ALL_SUCCESSFUL	
TARGET_INSTRUCTION	CREATE/LOAD_OR_REPLACE EXECUTING ABEND DETAILED	
D_O_TYPE	X'10200000'	
D_O_LOCAL_NAME	LOCAL.NAME	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3) DELETED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V1)	

Response message unit is built at C and sent to A:

EP_CHANGE_MGMT at C parses the CP-MSU, extracting the Request Change Control major vector. Based on the Install subvector encodings, a pre-test is performed successfully on the change file and it is installed in production non-removably. After installation, a post-test is successfully performed. To report installation, EP_CHANGE_MGMT builds a reply CP-MSU, identifies the target destination as NET1.A, chooses server parameters and specifies the agent unit-of-work correlator. Finally, it invokes FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-25).

As a result, the SNA/FS server encodes a new server object and SNA/DS builds and sends a message unit back to node A.

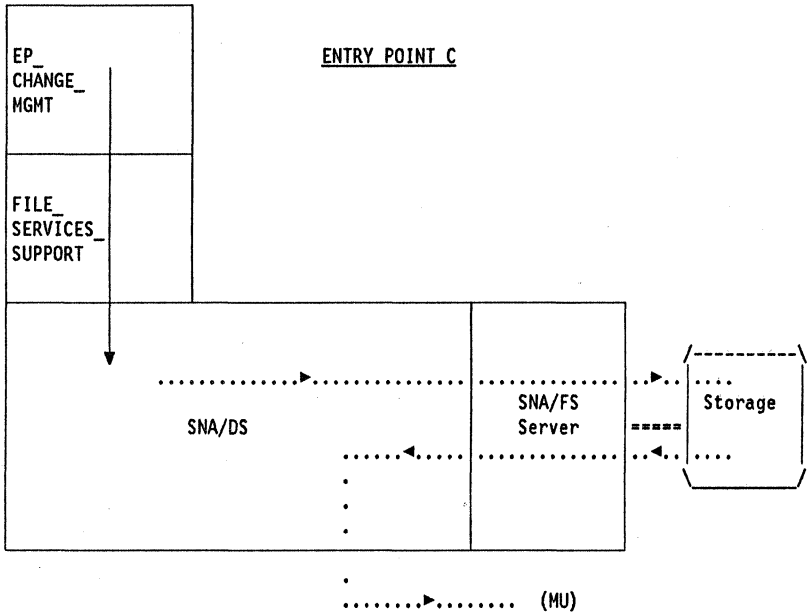


Figure 10-53. Destructive Send_and_Install (4 of 6). EP_CHANGE_MGMT at C invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-25. Send_Distribution Supplied Parameters		
Parameter Name	**Parameter Value**	**Parameter Meaning**
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	4321	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.A	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: CHANGE CONTROL MAJOR VECTOR CONSISTING OF: DATE/TIME SUBVECTOR REPORTING INSTALLATION SUBVECTOR REPORTED CHANGE NAME SUBVECTOR FS ACTION SUMMARY GDS	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
ENCODER_INSTRUCTION	ENCODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3) DELETED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V1)	

Response message unit is received at A:

SNA/DS at A receives the message unit and the SNA/FS server decodes the server object. Then SNA/DS invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-26).

FILE_SERVICES_SUPPORT then invokes CHANGE_MGMT_NETOP, passing it the CP-MSU, source destination (NET1.C), server parameters and agent unit-of-work correlator.

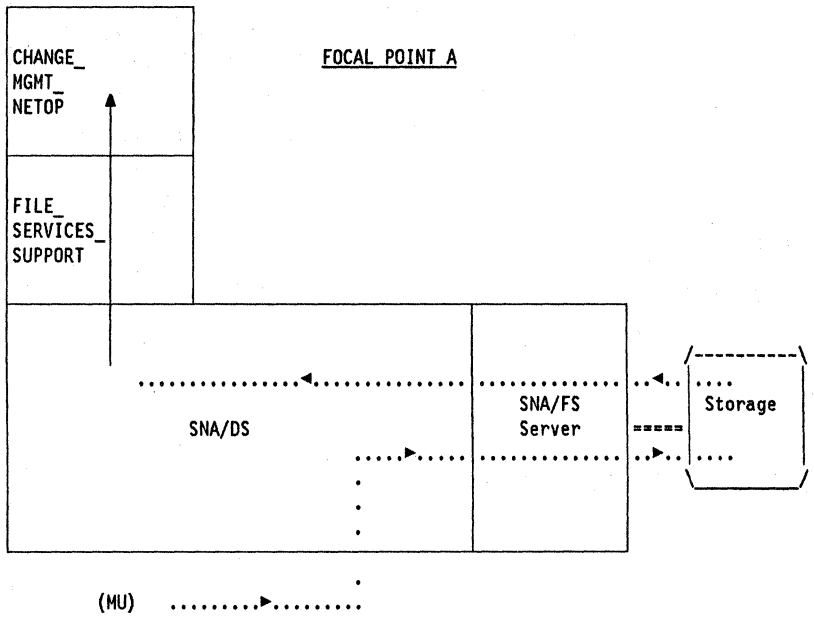


Figure 10-54. Destructive Send_and_Install (5 of 6). FILE_SERVICES_SUPPORT at A issues Receive_Distribution and is returned parameters. It then invokes CHANGE_MGMT_NETOP.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-26. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	4321	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: CHANGE CONTROL MAJOR VECTOR CONSISTING OF: DATE/TIME SUBVECTOR REPORTING INSTALLATION SUBVECTOR REPORTED CHANGE NAME SUBVECTOR FS ACTION SUMMARY GDS	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V3) DELETED_NAME (MCUST, 9135, NA, NET1, C, SYSTEM, V1)	

Network operator at A is issued response:

After parsing the reply CP-MSU and extracting data, CHANGE_MGMT_NETOP at A builds and issues, to the network operator, the Reporting_Installation protocol boundary verb for Change Management defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-27). Note that the Correlator value is the same as that originally provided by the network operator on the Send_and_Install verb.

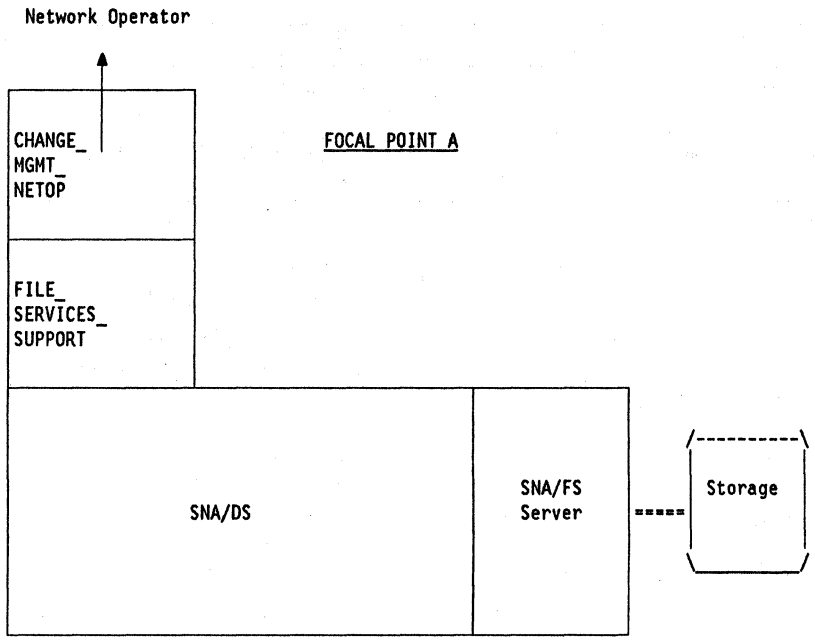


Figure 10-55. Destructive Send_and_Install (6 of 6). CHANGE_MGMT_NETOP at A issues Reporting_Installation verb to network operator.

The parameters returned on the Reporting_Installation verb to the network operator are:

Table 10-27. Reporting_Installation Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'723'	CORRELATES REPORT TO INITIAL REQUEST
TIME_STAMP	DATE/TIME OF INSTALLATION	FROM DATE/TIME SUBVECTOR IN RETURNED CP-MSU
TARGET_LOCATION	NET1.C	LOCATION WHERE FILE STORED
STORED_NAME	MCUST, 9135, NA, NET1, C, SYSTEM, V3	SNA/FS GLOBAL NAME OF STORED FILE
DELETED_NAME	MCUST, 9135, NA, NET1, C, SYSTEM, V1	SNA/FS GLOBAL NAME OF DELETED FILE
FS_ACTION_SUMMARY	ALL_SUCCESSFUL NO_BACKOUT_ATTEMPTED ABEND_NOT_APPLICABLE CONTINUE_NOT_ATTEMPTED	AS DEFINED BY SNA/FS
INSTALLATION_RESULTS INSTALLATION_STATUS WHEN_EFFECTIVE REPORTED_CHANGE_NAME_LIST CHANGE_FILE_NAME	 SUCCESSFUL ACTIVATION_REQUIRED MCUST, 9135, NA, NET1, C, SYSTEM, V3	 INSTALLATION SUCCESSFUL UPON ACTIVATION CHANGE FILE(S) INSTALLED
PRE_TEST_STATUS	SUCCESSFUL	TEST WAS SUCCESSFUL
POST_TEST_STATUS	SUCCESSFUL	TEST WAS SUCCESSFUL
AUTOMATIC_REMOVAL_RESULTS AUTOMATIC_REMOVAL_STATUS WHEN_EFFECTIVE	 NOT_ATTEMPTED NOT_APPLICABLE	 REMOVAL NOT ATTEMPTED EFFECTIVE TIME NOT APPLICABLE
AUTOMATIC_ACCEPTANCE_STATUS	NOT_ATTEMPTED	ACCEPTANCE NOT ATTEMPTED
REMOVABILITY_STATUS	INSTALLED_NONREMOVABLY	CHANGE FILE MAY NOT BE REMOVED
ACTIVATION_USE_STATUS	PRODUCTION	CHANGE FILE INSTALLED IN PRODUCTION

Failing INSTALL

Consider the two nodes in a SNA/DS network depicted below. Assume that both nodes happen to be in the same network (NET1).

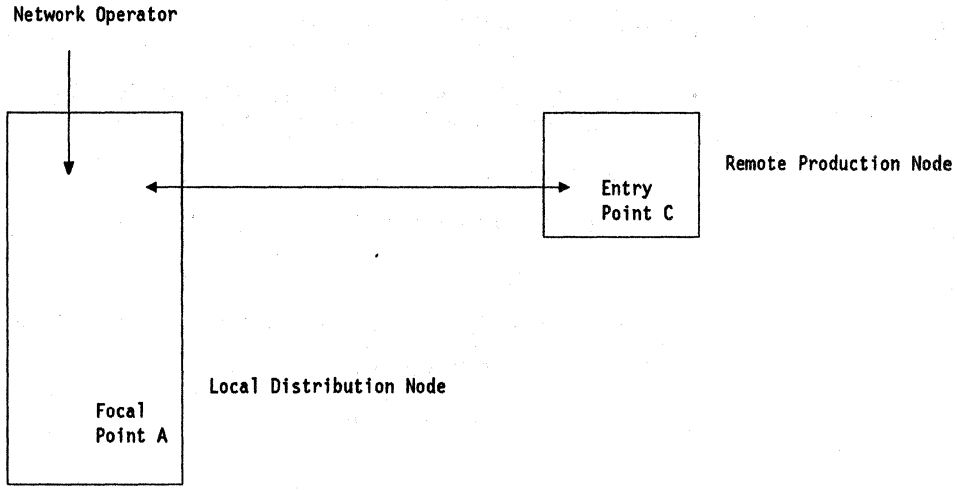


Figure 10-56. Sample Configuration

Given the configuration in Figure 10-56, in the example that follows, a microcode customizing data change file specifying system parameters with SNA/FS global name MCUST.9135.NA.NET1.C.LIB1.V2 and another microcode customizing data change file specifying printer parameters with SNA/FS global name MCUST.9135.NA.NET1.C.LIB4.V1 have been stored at, but not yet installed on, a remote production node (entry point C).

A network operator at A wishes to install both change files, as corequisites, at C. He chooses the following installation options:

- Trial installation
- Removability required
- Pre-test required
- Automatic Removal if test failure or installation failure
- Post-test not to be performed
- Automatic Acceptance prohibited

A discussion of these Install options can be found in "Change Control" on page 6-5.

Network operator at A issues request:

In the architected model, the network operator then assigns a sequence number and issues the Install Change Management request protocol boundary verb which is defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-28). However, a typical implementation will assign the sequence number automatically before issuing the verb.

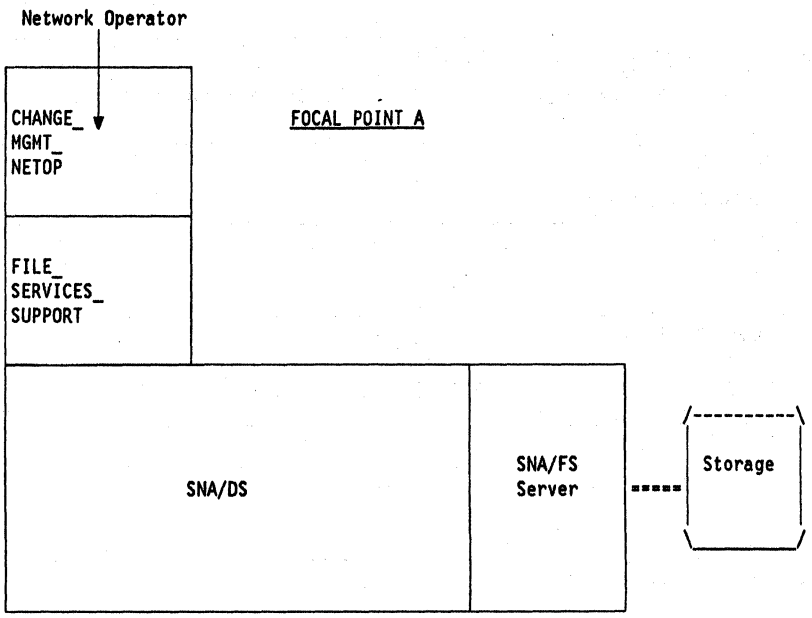


Figure 10-57. Failing Install (1 of 6). Network operator at A issues Send_and_Install verb to CHANGE_MGMT_NETOP.

The parameters supplied on the Install verb by the network operator are:

Table 10-28. Install Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'42'	CORRELATION VALUE FOR THIS UNIT OF WORK
STORED_NAME	MCUST, 9135, NA, NET1, C, LIB1, V2	SNA/FS GLOBAL NAME OF STORED CHANGE FILE
COREQUISITE_CHANGE_NAME_LIST CHANGE_FILE_NAME	MCUST, 9135, NA, NET1, C, LIB4, V1	SNA/FS GLOBAL NAME OF COREQUISITE CHANGE FILE(S)
TARGET_LIST TARGET_LOCATION	NET1.C	LOCATION(S) TO SEND CHANGE FILE
REMOVABILITY	YES	INSTALL FILE SO IT CAN BE PROCESSED BY SUBSEQUENT REMOVE PROTOCOL BOUNDARY VERB
AUTOMATIC_REMOVAL	YES	REMOVE CHANGE FILE IF INSTALLATION OR TEST FAILS
PRE_TEST	YES	PERFORM PRE-TEST
POST_TEST	NO	DO NOT PERFORM POST-TEST
AUTOMATIC_ACCEPTANCE	NO	DO NOT AUTOMATICALLY ACCEPT CHANGE FILE IF INSTALLATION AND TEST SUCCEED
ACTIVATION_USE	TRIAL	COMPONENTS ALTERED BY INSTALLATION WILL BE USED IN LATER TRIAL ACTIVATION

Request message unit is built at A and sent to C:

CHANGE_MGMT_NETOP at A builds the request CP-MSU, identifies the target destination as NET1.C, chooses server parameters and assigns an agent unit-of-work correlator. It then passes this data to FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-29).

As a result, the SNA/FS server encodes the server object. SNA/DS then builds and sends a message unit (MU) to remote node C.

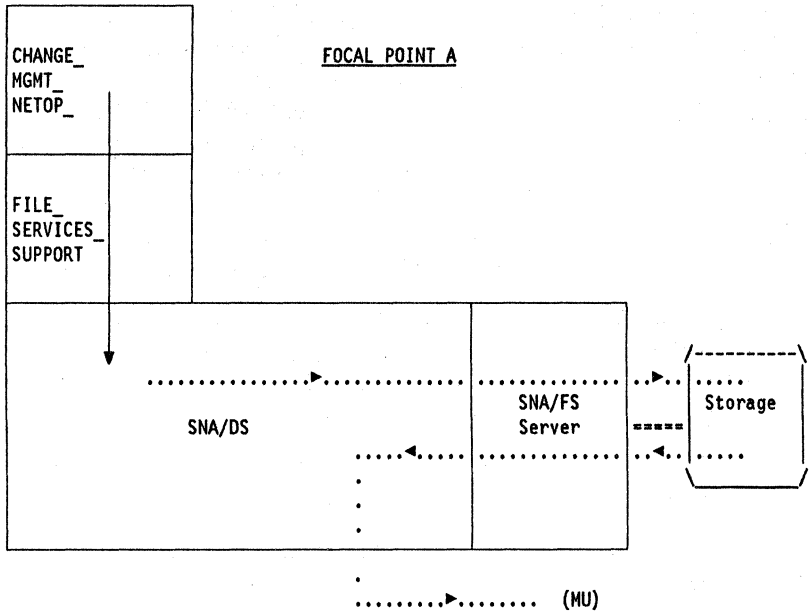


Figure 10-58. Failing Install (2 of 6). CHANGE_MGMT_NETOP at A invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-29. Send_Distribution Supplied Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	8765	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.C	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR COREQUISITE CHANGE SUBVECTOR	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
SERVER	X'24F0F0F0'	INDICATES SNA/FILE SERVICES
SPECIFIC_SERVER_INFO		AS DEFINED BY SNA/FS
ENCODER_INSTRUCTION	ENCODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, LIB1, V2)	

Request message unit is received at C:

SNA/DS at C receives the message unit and the SNA/FS server decodes the server object. Then SNA/DS invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-30).

FILE_SERVICES_SUPPORT then invokes EP_CHANGE_MGMT, passing it the CP-MSU, source destination (NET1.A), server parameters and agent unit-of-work correlator.

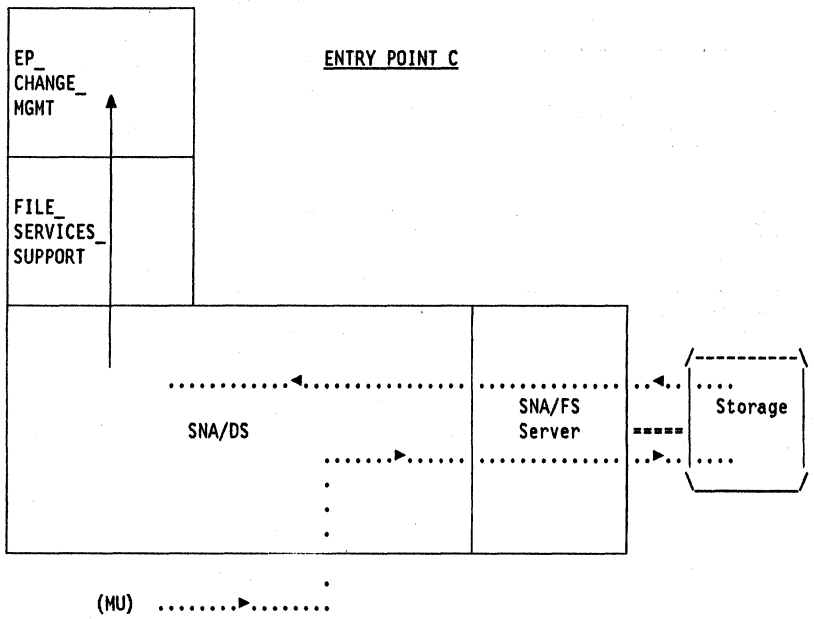


Figure 10-59. Failing Install (3 of 6). FILE_SERVICES_SUPPORT at C issues Receive_Distribution and is returned parameters. It then invokes EP_CHANGE_MGMT.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-30. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.A	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	8765	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: REQUEST CHANGE CONTROL MAJOR VECTOR CONSISTING OF: INSTALL SUBVECTOR COREQUISITE CHANGE SUBVECTOR	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED
SPECIFIC_SERVER_REPORT		AS DEFINED BY SNA/FS
DECODER_INSTRUCTION	DECODE_ONLY ABEND ONLY_IF_EXCEPTIONS	
D_O_TYPE	X'10200000'	
D_O_CANONICAL_NAME	STORED_NAME (MCUST, 9135, NA, NET1, C, LIB1, V2)	

Response message unit is built at C and sent to A:

EP_CHANGE_MGMT at C parses the CP-MSU, extracting the Request Change Control major vector. A new microcode change file with SNA/FS global name MCODE.9135.NA.FUNCTEC.A38069.CONTROL required by the new customizing data files, is discovered to be missing. This requirement is described in the coverletter for the maintenance but was missed by the network operator.

To report the error, EP_CHANGE_MGMT builds a reply CP-MSU that includes an SNA Condition Report as well as a Change Management Reply major vector containing installation status, file identification and additional error related detailed data ((5201,01), (52aa,02) (52ff,01)). EP_CHANGE_MGMT identifies the report target destination as NET1.A and specifies the agent unit-of-work correlator (note that no SNA/FS server parameters are specified in this case). Finally, it invokes FILE_SERVICES_SUPPORT which issues a Send_Distribution protocol boundary verb for SNA/DS (see Table 10-31).

As a result, SNA/DS builds and sends a message unit back to node A.

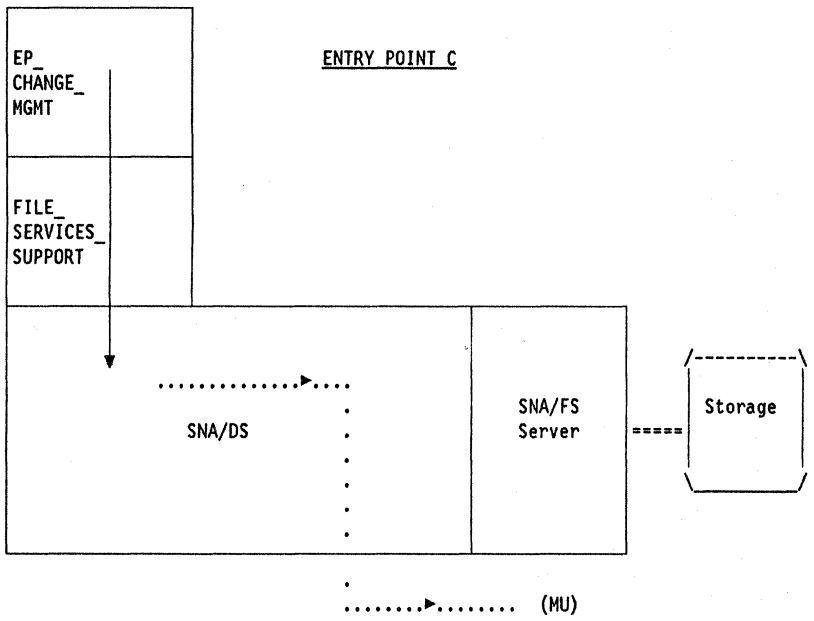


Figure 10-60. Failing Install (4 of 6). EP_CHANGE_MGMT at C invokes FILE_SERVICES_SUPPORT which issues Send_Distribution.

The parameters supplied on the SNA/DS Send_Distribution verb are:

Table 10-31. Send_Distribution Supplied Parameters		
Parameter Name	**Parameter Value**	**Parameter Meaning**
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	2109	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
DESTINATION	NET1.A	DSU NAME OF RECIPIENT
DEST_AGENT	X'23F0F0F0'	INDICATES SNA/MS
AGENT_OBJECT	CP-MSU CONSISTING OF: CHANGE CONTROL MAJOR VECTOR CONSISTING OF: DATE/TIME SUBVECTOR REPORTING INSTALLATION SUBVECTOR REPORTED CHANGE NAME SUBVECTOR FOR FIRST CHANGE FILE REPORTED CHANGE NAME SUBVECTOR FOR COREQ CHANGE FILE DETAILED DATA SUBVECTOR SNA CONDITION REPORT GDS VARIABLE IDENTIFYING MISSING CHANGE FILE	SEE *SNA Formats, GA27-3136*
SERVICE_PARMS		
PRIORITY	DATA8	USE AT LEAST THIS PRIORITY
PROTECTION	LEVEL2	SAFE STORE MUST BE USED
CAPACITY	16MEG	HANDLE AT LEAST 16M OBJECT
SECURITY	LEVEL2	SECURITY REQUIRED
ACCEPT_DELAY	INDEFINITE	INDEFINITE DELAY ACCEPTABLE
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
INTEGRITY	HIGH	USE CONFIRMATION PROTOCOLS
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO

Response message unit is received at A:

SNA/DS at A receives the message unit and invokes FILE_SERVICES_SUPPORT, which issues the Receive_Distribution SNA/DS protocol boundary verb and is returned parameters (see Table 10-32).

FILE_SERVICES_SUPPORT then invokes CHANGE_MGMT_NETOP, passing it the CP-MSU, source destination (NET1.C) and agent unit-of-work correlator.

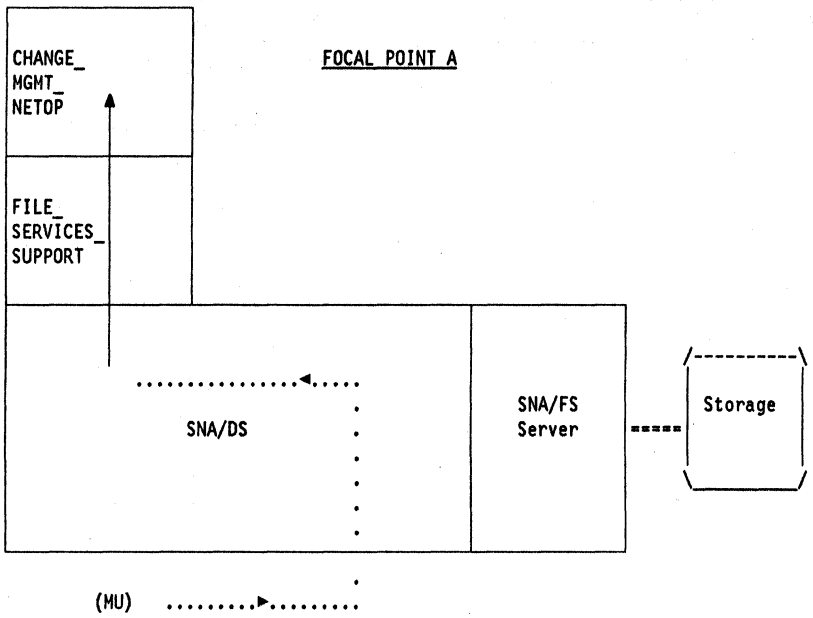


Figure 10-61. Failing Install (5 of 6). FILE_SERVICES_SUPPORT at A issues Receive_Distribution and is returned parameters. It then invokes CHANGE_MGMT_NETOP.

The parameters returned on the SNA/DS Receive_Distribution verb are:

Table 10-32. Receive_Distribution Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
DISTRIBUTION_ID		
ORIGIN_DSU	NET1.C	DSU NAME OF DIST ORIGIN
ORIGIN_AGENT	X'23F0F0F0'	INDICATES SNA/MS
ORIGIN_SEQNO	2109	SEQUENCE NUMBER
AGENT_CORREL	AGENT CORRELATION VALUE	SEE *SNA Formats, GA27-3136*
AGENT_OBJECT	CP-MSU CONSISTING OF: CHANGE CONTROL MAJOR VECTOR CONSISTING OF: DATE/TIME SUBVECTOR REPORTING INSTALLATION SUBVECTOR REPORTED CHANGE NAME SUBVECTOR FOR FIRST CHANGE FILE REPORTED CHANGE NAME SUBVECTOR FOR COREQ CHANGE FILE DETAILED DATA SUBVECTOR SNA CONDITION REPORT GDS VARIABLE IDENTIFYING MISSING CHANGE FILE	SEE *SNA Formats, GA27-3136*
REPORTING_REQUESTED		
EXCEPTION	YES	REQUEST EXCEPTION REPORT
REPORT_TO_DSU	NET1.A	DSU TO SEND DIST REPORTS TO
DISTRIBUTION_TIME	HH.MM.SS.HH.GMT	GMT TIME AT WHICH DISTRIBUTION ORIGINATED

Network operator at A is issued response:

After parsing the reply CP-MSU and extracting data, CHANGE_MGMT_NETOP at A builds and issues, to the network operator, the Reporting_Installation protocol boundary verb for Change Management defined in Appendix B, "Management Services Protocol Boundary Verbs" on page B-1 (see Table 10-33). Note that the Correlator value is the same as that originally provided by the network operator on the Send_and_Install verb.

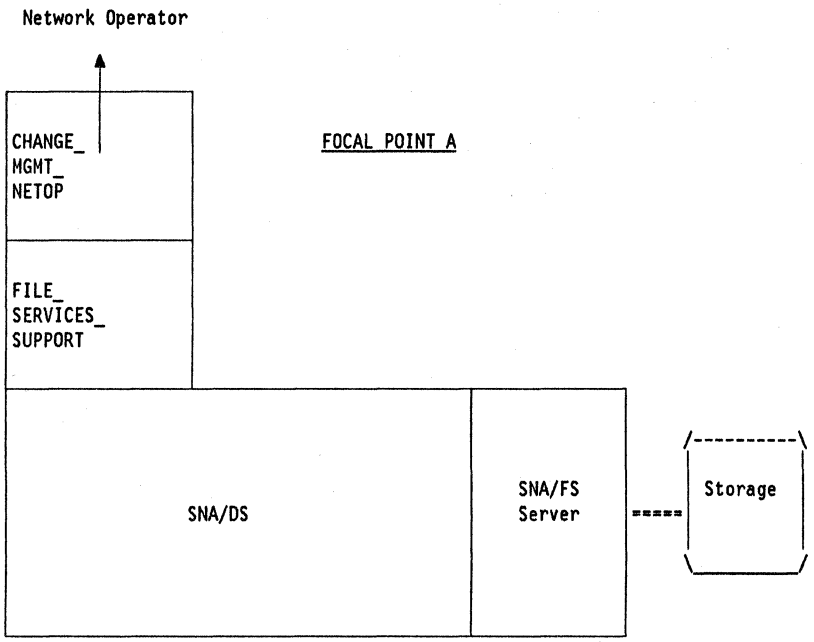


Figure 10-62. Failing Install (6 of 6). CHANGE_MGMT_NETOP at A issues Reporting_Installation verb to network operator.

The parameters returned on the Reporting_Installation verb to the network operator are:

Table 10-33. Reporting_Installation Returned Parameters

Parameter Name	Parameter Value	Parameter Meaning
CORRELATOR	X'42'	CORRELATES REPORT TO INITIAL REQUEST
TIME_STAMP	DATE/TIME OF INSTALLATION	FROM DATE/TIME SUBVECTOR IN RETURNED CP-MSU
TARGET_LOCATION	NET1.C	LOCATION WHERE FILE STORED
INSTALLATION_RESULTS		
INSTALLATION_STATUS	NOT_ATTEMPTED	INSTALL NOT ATTEMPTED
WHEN_EFFECTIVE	NOT_APPLICABLE	TIME NOT APPLICABLE
REPORTED_CHANGE_NAME_LIST		CHANGE FILE(S) BEING REPORTED ON
CHANGE_FILE_NAME	MCUST, 9135, NA, NET1, C, LIB1, V2	
CHANGE_FILE_NAME	MCUST, 9135, NA, NET1, C, LIB4, V1	
PRE_TEST_STATUS	NOT_ATTEMPTED	TEST NOT ATTEMPTED
POST_TEST_STATUS	NOT_ATTEMPTED	TEST NOT ATTEMPTED
AUTOMATIC_REMOVAL_RESULTS		
AUTOMATIC_REMOVAL_STATUS	NOT_ATTEMPTED	REMOVAL NOT ATTEMPTED
WHEN_EFFECTIVE	NOT_APPLICABLE	EFFECTIVE TIME NOT APPLICABLE
AUTOMATIC_ACCEPTANCE_STATUS	NOT_ATTEMPTED	ACCEPTANCE NOT ATTEMPTED
REMOVABILITY_STATUS	NOT_INSTALLED	INSTALLATION WAS NOT SUCCESSFUL
ACTIVATION_USE_STATUS	NOT_INSTALLED	INSTALLATION WAS NOT SUCCESSFUL
DETAILED_DATA	(5201,01), (52AA,02), (52FF,01)	ERROR CODES UNIQUE TO ENTRY POINT IMPLEMENTATION
SNA_CONDITION_REPORT		
SNA_REPORT_CODE	X'08380012'	MISSING COREQUISITE CODE
REPORTED_ON_TOKEN_STRING	MCODE, 9135, NA, FUNCTEC, A38069, CONTROL	SNA/FS GLOBAL NAME OF MISSING COREQUISITE FILE

EP_COMMON_OPERATIONS_SERVICES Function Set

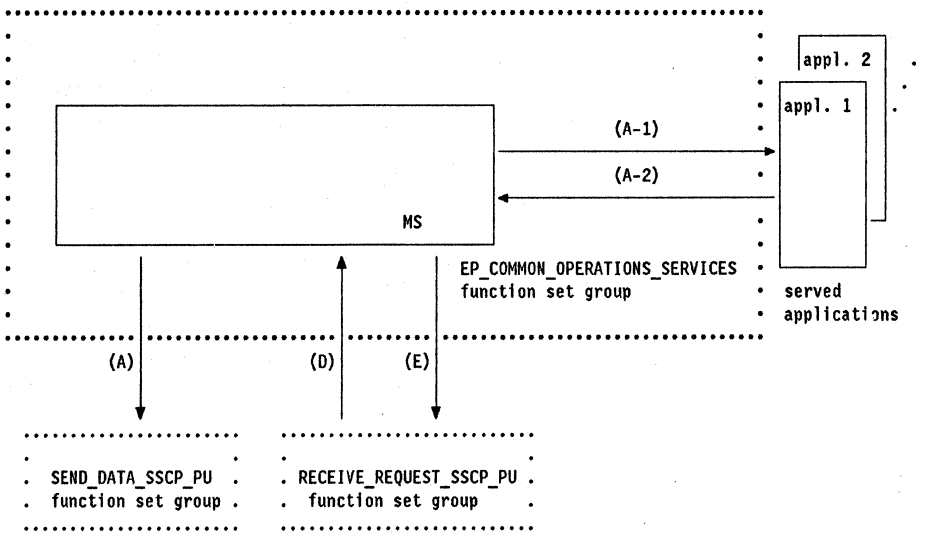


Figure 10-63. EP_COMMON_OPERATIONS_SERVICES Function Set Group

The EP_COMMON_OPERATIONS_SERVICES function set provides the capability to receive common operations services commands and route them to specified network management applications located at its node, to return to the command senders replies to these commands generated by the applications, and to route messages from the applications to specified network operators.

Refer to Figure 10-63 throughout the discussion of the EP_COMMON_OPERATIONS_SERVICES function set.

Protocol Boundaries with Components Outside EP_COMMON_OPERATIONS_SERVICES

- Input:
  - Internal MS protocol boundary D (NMVT Received)

    The EP_COMMON_OPERATIONS_SERVICES function set group receives requests from the RECEIVE_REQUEST_SSCP_PU function set group to process incoming NMVTs. It is passed the NMVT and the name of the CP from which it was received. The details of this protocol boundary are described in "Protocol Boundary D - NMVT Received" on page 8-19.
  - MS application protocol boundary A-2

    The EP_COMMON_OPERATIONS_SERVICES function set group receives replies to earlier commands, as well as unsolicited messages to operators, from served network management applications in its node. It is passed the one or more completed MS major vectors, and, in the case of replies, the correlator that it passed to the application with the request. The details of this protocol boundary are described in Table 10-36 on page 10-144.

- Output:
  - Internal MS protocol boundary A (Send NMVT)

    The EP_COMMON_OPERATIONS_SERVICES function set group requests the SEND_DATA_SSCP_PU function set group to send an NMVT RU on the SSCP-PU session with its controlling CP. The output data consists of the complete NMVT and the address of the control point to which it is to be sent. The details of this protocol boundary are described in "Protocol Boundary A - Send NMVT" on page 8-18.

  - Internal MS protocol boundary E (Send NMVT Response)

    The EP_COMMON_OPERATIONS_SERVICES function set group requests the SEND_DATA_SSCP_PU function set group to send an NMVT response RU on an SSCP-PU session with a specified CP. The output data consists of the CP address and sense data. The details of this protocol boundary are described in "Protocol Boundary E - Send NMVT Response" on page 8-19.

  - MS application protocol boundary A-1

    The EP_COMMON_OPERATIONS_SERVICES function set group passes commands to served network management applications in its node. For each command it passes a complete MS major vector, together with a correlator that the application will retain to pass back with the reply. The details of this protocol boundary are described in Table 10-35 on page 10-143.

Prerequisite Function Sets

See "Role Requirements for Management Services Components" on page 8-21 for information on the relationships between this function set and other function sets.

Overview of Subsets

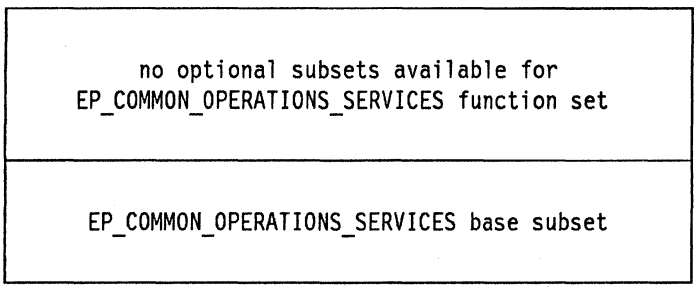


Figure 10-64. Base and Optional Subsets of EP_COMMON_OPERATIONS_SERVICES Function Set

EP_COMMON_OPERATIONS_SERVICES Base Subset

Functions Provided

The EP_COMMON_OPERATIONS_SERVICES base subset provides the capability to support communication between network operators and served network management applicatons, as described in "Common Operations Services for Resource Control" on page 7-3.

Formats Supported

The EP_COMMON_OPERATIONS_SERVICES base subset supports the receiving of an NMVT containing one of four management services major vectors:

- Execute Command (X'8061')
- Analyze Status (X'8062')
- Query Resource Data (X'8063')
- Test Resource (X'8064')

The base subset passes each of these major vectors to a served network management application, which is named in a Name List (X'06') subvector within the major vector.

The EP_COMMON_OPERATIONS_SERVICES base subset also supports receipt, from served network management applications, of the combinations of management services major vectors shown in Table 10-34 on page 10-141.

EP_COMMON_OPERATIONS_SERVICES

Table 10-34. Management Services Major Vectors Supported by	
Major Vectors	**Number**
Reply to Execute Command (X'0061') Text Data (X'1300') Structured Data (X'1307') Transparent Coded Datastream (X'1309')	1 0–1 0–1 0–1
Reply to Analyze Status (X'0062') Begin Data Parameters (X'130A') Structured Data (X'1307') End Parameter Data (X'130B')	1 1 0-n 1
Reply to Query Resource Data (X'0063') Begin Data Parameters (X'130A') Structured Data (X'1307') End Parameter Data (X'130B')	1 1 1-n 1
Reply to Test Resource (X'0064') Begin Data Parameters (X'130A') Structured Data (X'1307') End Parameter Data (X'130B')	1 1 0-n 1
Send Message to Operator (X'006F') Text Data (X'1300') Structured Data (X'1307') Transparent Coded Datastream (X'1309')	1 0–1 0–1 0–1

The major vectors having keys beginning with X'13' are MS *parameter* major vectors; see "Parameter Major Vectors" on page 7-4 for a discussion of MS major vectors of this type.

Electives

The only elective available to this subset is the reporting of requests that cannot be processed by sending sense data in either a negative response or reply.

Implementation Requirements

The requirements for implementing the EP_COMMON_OPERATIONS_SERVICES base subset are described by a model consisting of a subset of Management Services.

A Subset of Management Services:

- Receiving common operations services requests

  Upon request by the RECEIVE_REQUEST_SSCP_PU function set group (via protocol boundary D), does the following:

  – Parses the request for validity and requests the RECEIVE_REQUEST_SSCP_PU function set group (via protocol boundary E), to send a positive or negative response.

- If the request parsed correctly, passes two items to the network management application identified in the request: the major vector, and a correlator defined by EP_COMMON_OPERATIONS_SERVICES (via MS application protocol boundary A-1).

- Sending common operations services data

  Upon request by a served network management application, does the following:

  - Receives two or more completed management services major vectors from the application. For a reply, the application also passes to EP_COMMON_OPERATIONS_SERVICES the correlator that it received with the request. (Via MS application protocol boundary A-2).
  - For a reply, uses the correlator to retrieve the PRID saved from the request.
  - Using the major vectors and the PRID, builds a complete NMVT.
  - Passes the NMVT to the SEND_DATA_SSCP_PU function set group via protocol boundary A.

Receiving Common Operations Services Requests:

This process is started by the process described in "Receiving NMVTs" on page 9-18 (RECEIVE_REQUEST_SSCP_PU function set group), when the latter has identified one of four major vector keys in a request NMVT:

- X'8061' (Execute Command)
- X'8062' (Analyze Status)
- X'8063' (Query Resource Data)
- X'8064' (Test Resource)

The first task of this process is to parse the common operations services major vector; in particular, it searches for the Destination Application Name (X'50') subfield within the Name List (X'06') subvector. If it cannot locate this subfield, or if the network management application identified in the subfield is not known to it, then this process starts the process described in "Sending NMVT Responses" on page 9-19 to send the appropriate -RSP. The sense data that can be sent by PUMS at this time are detailed in Table 10-37 on page 10-152 and Table 10-42 on page 10-159. If no condition requiring a -RSP is found, then this process starts the process described in "Sending NMVT Responses" (via protocol boundary E) to send a +RSP,

Next this process creates a correlator to pass to the served application when it passes it the major vector from the request. EP_COMMON_OPERATIONS_SERVICES retains an association between this correlator and the PRID from the request. The purpose of this private correlator, used only on the EP_COMMON_OPERATIONS_SERVICES application protocol boundaries A-1 and A-2, is to shield served applications from differences in, and changes to, the manner in which management services data is transported between nodes. Since knowledge of request/reply correlation is used by transport components as well as by the applications using these components to communicate, correlators, such as the PRID, that actually flow between nodes tend to be transport-specific. By

mapping the PRID to its own private correlator, EP_COMMON_OPERATIONS_SERVICES insures that the network management applications it serves will not be affected by changes to the underlying management services transport.

This process now passes the major vector from the request, along with its private correlator, to the application, via MS application protocol boundary A-1. See Table 10-35 for details of this protocol boundary.

After passing the major vector and correlator to the designated application, this process terminates.

Table 10-35. MS Application Protocol Boundary A-1	
Origin	EP_COMMON_OPERATIONS_SERVICES
Destination	The served network management application identified in the common operations services request.
Data Content	1. A pointer to the common operations services major vector received in an NMVT 2. A private token for request/reply correlation
Initiates	The network management applicatoin identified in the common operations services request.

Sending Common Operations Services Data:

This process is started by a served network management application when the application has common operations services data to send. The application passes to EP_COMMON_OPERATIONS_SERVICES (via protocol boundary A-2), a number of management services major vectors, and, in the case of a reply, the correlator that it received with the request. See Table 10-36 on page 10-144 for a description of MS application protocol boundary A-2.

Table 10-34 on page 10-141 summarizes the combinations of major vectors that a network management application can pass to EP_COMMON_OPERATIONS_SERVICES. "Common Operations Services for Resource Control" on page 7-3 describes more fully which major vectors are to be sent by an application in reply to each of the common operations services commands, and which may be sent unsolicited. EP_COMMON_OPERATIONS_SERVICES, however, makes no attempt to enforce these rules; it envelopes and sends the major vectors it receives, without examining their contents.

For a reply, this process uses the correlator passed to it with the major vectors to retrieve the PRID from the original request.

Using the PRID that it has just retrieved, this process now constructs a complete NMVT enveloping the major vectors.

This process starts the process described in "Sending NMVTs" on page 9-14 (SEND_DATA_SSCP_PU function set group) via protocol boundary A, to send the NMVT. "Protocol Boundary A - Send NMVT" on page 8-18 shows the items

passed when starting that process. This process places the following values in these items:

Item 1: The NMVT it has constructed

Item 2: The address of the node's controlling control point (present only on replies)

After this process has started the "Sending NMVTs" process, it terminates.

Table 10-36. MS Application Protocol Boundary A-2	
Origin	A served network management application
Destination	EP_COMMON_OPERATIONS_SERVICES
Data Content	1. A pointer to a set of common operations services major vectors that the served application has built 2. The private token for request/reply correlation that was passed to the application with the request **Note:** Item 2 is passed only on replies.
Initiates	Process described in "Sending Common Operations Services Data" on page 10-143

Common Operations Services Commands, Replies, and Unsolicited Traffic

The architecture for common operations services provides for the transfer of four commands from a network operator to a specified network management application, and the return of the corresponding replies from the application to the operator. It also provides for the transfer of a message from an application to a specified operator. The remainder of this section provides a brief description of how each of these five functions is implemented. Before proceeding, the reader should review "Parameter Major Vectors" on page 7-4.

Execute Command

- *Execute Command:*

  This command is entered by a network operator to request that a text command be sent; the command itself is not interpreted by the common operations services component in the control point. In this case the Execute Command (X'8061') major vector is built, with the following subvectors:

  - X'06' The Name List subvector, with a Destination Application Name (X'50') subfield, is built to transport the operator-specified name of the destination application to which the command is to be routed.
  - X'31' The Self-Defining Text Message subvector is built as follows, to transport the command:
    - Coded Character Set ID (X'02') subfield: always set to X'028001F4', i.e., Coded Graphic Character Set 00640–00500
    - Text Message (X'30') subfield: this subfield contains, as text, the command to be transported.

- *Reply to Execute Command:*

  A positive reply to the Execute Command major vector consists of an empty Reply to Execute Command (X'0061') major vector, followed by one of three parameter major vectors:

  - Text Data (X'1300')
  - Structured Data (X'1307')
  - Transparent Coded Datastream (X'1309')

  A negative reply to the Execute Command major vector consists of a Reply to Execute Command major vector containing a Sense Data (X'7D') subvector indicating why the command failed. No parameter major vector is present in this case.

Analyze Status

- *Analyze Status:*

  This command is entered by a network operator to request that information about one or more specified resources be gathered and analyzed. The reply to this command reports the joint state of all the specified resources. In this case the Analyze Status (X'8062') major vector is built. This major vector contains a single subvector, the Name List (X'06') subvector, which is built as follows:

  - The Destination Application Name (X'50') subfield is always built, to carry the operator-specified name of the destination application to which the command is to be routed.
  - The Associated Resource Name List (X'01') subfield is always built.

    Note: This contents of this subfield are not interpreted either by the common operations services component in the control point or by EP_COMMON_OPERATIONS_SERVICES; its entries are specified by the network operator, and interpreted by the served application to which EP_COMMON_OPERATIONS_SERVICES routes the request. There are three mutually incompatible ways in which the subfield is currently used:

    - The entries in the list may identify individual discrete resources to which the command carried in the request is to be applied.
    - The entries in the list may identify entities that delimit a *range* of resources to which the command is to be applied. In the case of a complex link, for example, the subfield might contain only the names of the resources at the two ends of the link, with the understanding that the request applies not only to these two specific resources, but also to all the other resources between them on the complex link.
    - Some or all of the entries in the list may serve as nicknames that index lists of actual resource names known to the served application that processes the request.

Obviously the network operator initiating a common operations services request must understand in which of these three ways the served application receiving the request will interpret the Associated Resource Name List subfield.

- *Reply to Analyze Status:*

  A positive reply to the Analyze Status major vector consists of an empty Reply to Analyze Status (X'0062') major vector, followed by a list of parameter major vectors:

  - One Begin Data Parameters (X'130A') major vector
  - Zero or more Structured Data (X'1307') major vectors
  - One End Parameter Data (X'130B') major vector.

  Each Structured Data major vector present in the reply contains data for a single resource.

  A negative reply to the Analyze Status major vector consists of a Reply to Analyze Status major vector containing a Sense Data (X'7D') subvector indicating why the command failed. No parameter major vector is present in this case.

Query Resource Data

- *Query Resource Data:*

  This command is entered by a network operator to request that information be gathered and returned from one or more specified resources. In this case the Query Resource Data (X'8063') major vector is built, with the following subvector:

  - X'06' The Name List subvector is built in exactly the same way it was for the Analyze Status (X'8062') major vector.

- *Reply to Query Resource Data:* The replies to the Query Resource Data major vector have exactly the same structure as those to the Analyze Status major vector, except that a positive reply always includes at least one Structured Data (X'1307') major vector.

Test Resource

- *Test Resource:*

  This command is entered by a network operator to request that one or more specified resources be tested. The reply to this command reports the state of each resource. In this case the Test Resource (X'8064') major vector is built, with the following subvectors:

  - X'06' The Name List subvector is built in exactly the same way it was for the Analyze Status (X'8062') major vector.
  - X'81' The Test Setup Data subvector is built based on information provided by the operator.

- *Reply to Test Resource:*

A positive reply to the Test Resource major vector consists of an Reply to Test Resource (X'0064') major vector, followed by a list of parameter major vectors:

- One Begin Data Parameters (X'130A') major vector
- Zero or more Structured Data (X'1307') major vectors
- One End Parameter Data (X'130B') major vector.

The Reply to Test Resource major vector contains a Test Result Data (X'81') subvector reporting the results of the test that was requested. Each Structured Data major vector present in the reply contains data for a single resource.

A negative reply to the Test Resource major vector consists of a Reply to Test Resource major vector containing a Sense Data (X'7D') subvector

Send Message to Operator

- *Send Message to Operator:* This function is invoked by a network management application, to send a message to a network operator. In this case a Send Message to Operator (X'006F') major vector is built; the included Name List (X'06') carries, as text, the identity of the operator to which the message is to be delivered. Following the Send Message to Operator major vector is one of three parameter major vectors:
  - Text Data (X'1300')
  - Structured Data (X'1307')
  - Transparent Coded Datastream (X'1309')

Common EP_XXXX Functions

Building the Date/Time (X'01') and Relative Time (X'42') Subvectors

There are four electives available to implementations for time stamping management services major vectors:

- The choice of providing either a Date/Time (X'01') subvector or Relative Time (X'42') subvector in the major vector.
- If the implementation elects to provide a Relative Time subvector, it has a further choice of the increment of time measurement.
- If the implementation elects to provide a Date/Time subvector, it has a further choice of providing an indication of time at a greater precision than seconds, e.g., milliseconds or microseconds.
- If the implementation elects to provide a Date/Time subvector, it has a further choice of providing the offset between the local time sent in the Date/Time subvector and Greenwich Mean Time.

Depending on the capabilities of the node, the elective for either date/time or relative time will be selected.

- If the elective for providing a date/time subvector is selected, this process constructs a Date/Time (X'01') subvector. If the elective for providing time at a finer granularity than seconds is selected, the binary value providing finer granularity than seconds is placed in the subvector field denoted 'Optional extension of time'. If the elective for providing GMT offset is selected, the subfield containing this offset is included.
- If the elective for providing a relative time subvector is selected, this process constructs a Relative Time (X'42') subvector. The time increment of measure is specified according to the elective selected. Any of the units of measure listed in the definition of the Relative Time subvector are equally acceptable.

Building the SNA Address List (X'04') Subvector

A process building an SNA Address List (X'04') subvector starts with either a single local address, or a pair of local addresses. In the latter case one address is for a local resource, and the other is for its session partner. The process proceeds as follows:

- The Address format field in the subvector is filled in:
  - For a single local address, the value X'00' (one or more single local addresses) is placed in this field.
  - For a pair of local addresses, the value X'40' (one or more pairs of session partner local addresses, each pair identifying a session) is inserted.
- The address of the local resource is inserted into the subvector.

- If a session partner address is present, it is inserted in the subvector *after* the resource address.
- The Address Count field in the subvector is filled in: X'01' for a single address, X'02' for an address pair.

Building the Product Set ID (X'10') Subvector

A process building a Product Set ID (X'10') subvector performs the sequence of steps described below. The Product Set ID subvector contains one or more Product Identifier (X'11') MS subvectors. The following discussion provides details on how these subvectors are constructed.

In certain cases the EP_ALERT function set group constructs two Product Set IDs, one identifying the node sending the Alert, the other identifying the origin of the Alert condition. For the first of these two Product Set IDs, the term "resource" in the discussion below refers to the PU sending the Alert. For the second Product Set ID it refers to the the origin of the Alert condition.

X'10': This serves as an "envelope" for either:

- One X'11' subvector for a resource that implements all of its product set in hardware; or
- Two or more X'11' subvectors for a resource that implements part or all of its product set in software. One subvector identifies the hardware, the others identify the software. If the resource is comprised of more than one software product, a X'11' subvector is built for each product.

X'11'(hardware): The Product Identifier subvector identifying the hardware is constructed containing the following subfields:

- The Hardware Product Identifier (X'00') subfield is constructed. The machine type and serial number (or repair ID number) are required from all products; the model number is also required if it is applicable. The format type is determined as follows:
  - One factor in selecting the format type is the mechanism used for identifying instances of the product. Individual instances of a product may be identified by serial number, by repair ID number, or by both. All products are assigned serial numbers; some are assigned repair ID numbers as well.

    The repair ID number serves to identify a product instance for such purposes as service contracts. When a customer first acquires a product instance, the repair ID number and the serial number are typically the same. If, however, the original instance of the product fails and must be exchanged for a new one, the original repair ID number is carried over to the new instance; thus the repair ID number and the serial number for the replacement product are no longer identical.

    For products having numbers of both kinds, the one that is *more visible externally on the product* is the one sent in this subfield. For serial numbers, one of the X'10', X'11', or X'12' formats is

chosen. For repair ID numbers, one of the X'20', X'21', or X'22' formats is chosen.

- The other factor determining the format used is the amount of information needed to identify a product instance uniquely.
  - The X'11' (or X'21') format is used when model number, machine type, and serial number (or repair ID number) are required to uniquely identify a product instance.
  - The X'12' (or X'22') format is used when model number does not assist in uniquely identifying a product instance, but is provided for the purpose of additional information only.
  - The X'10' (or X'20') format is used when machine type and a serial number (repair ID number) are required to uniquely identify a product instance and a model number is not provided.

- The Microcode EC Level (X'0B') subfield is constructed; inclusion of this subfield is optional for a product.
- The Hardware Product Common Name (X'0E') subfield is constructed; inclusion of this subfield is optional for a product.
- The Emulated Product Identifier (X'01') subfield is constructed when a product is emulating another hardware product.

X'11'(software): If any of the resource's product set is implemented in software, the Product Identifier subvector identifying the software is constructed containing the following subfields:

- The Software Product Serviceable Component Identifier (X'02') subfield is constructed if the product is supported by the IBM National Service Division (NSD).
- The Software Product Program Number (X'08') and Software Product Common Level (X'04') subfields are constructed for all software products that are not assigned a serviceable component identifier (are not supported by the National Service Division).
- The Software Product Common Name (X'06') subfield is always constructed. It provides a user-friendly means of identifying a software product.
- The Software Product Customization Identifier (X'07') or Software Product Customization Date and Time (X'09') subfield is constructed if the software product is customer modifiable. It provides unique identification of a particular set of instructions when multiple copies of the same software product exists in a system.

Building a Management Services Major Vector

A process constructs a major vector by enveloping the subvectors it has constructed earlier. Figure 2-2 on page 2-5 shows the relationship between the 2-byte major vector length, the 2-byte major vector key, and the subvectors. There are four constraints on the placement of the subvectors within a major vector:

One constraint applies to all major vectors:

- If it is present, the SNA Address List (X'04') subvector must be placed first.

Three additional constraints are unique to the Alert (X'0000') major vector:

- Any Detail Qualifier (X'A0' or X'A1') subvectors must be placed in the major vector by EP_ALERT in the same order that they were received from the LMS. This is done to insure that the qualifiers can be displayed to the network operator at the control point in the same order they were passed from the Alerting component to EP_ALERT.
- If both a Hierarchy Name List (X'03') and a Hierarchy/Resource List (X'05') subvector are included, the Hierarchy/Resource List subvector is placed ahead of the Hierarchy Name List subvector in the major vector.
- If two Product Set ID (X'10') subvectors are included, the one identifying the PU sending the Alert appears before the one identifying the origin of the Alert condition.

Building an NMVT

A process building an NMVT proceeds as follows:

Note: Any fields not described below are set to 0.

- The NS Header (X'41038D') is placed in bytes 0–2.
- The PRID, bits 4–15 of bytes 5–6, is filled in:
  - For an unsolicited NMVT, the value X'000' is inserted.
  - For a reply NMVT, the PRID value from the request is inserted.
- The flags (byte 7) are set:
  - The solicitation indicator is set to 0 for an unsolicited NMVT, 1 for a reply.
  - For unsolicited NMVTs, the sequence field is filled with B'00' (only NMVT for this PRID).
  - For a reply, the process building the NMVT places the appropriate value in the sequence field. Different processes use different techniques for determining what value belongs in this field. EP_COMMON_OPERATIONS_SERVICES, for example, supports only a single reply per request, so it always specifies B'00' (only reply for this PRID). EP_QPI, on the other hand, is passed an explicit indication with each reply by the physical resources manager LMS.
  - The SNA Address List indicator is set to either 0 or 1, depending upon whether an SNA Address List subvector was included in the major vector.

- The major vector is inserted, beginning in byte 8.

Parsing of NMVTs

The following table describes a parsing sequence that is common to all NMVT requests. This level of parsing is performed by the RECEIVE_REQUEST_SSCP_PU function set, before it passes the NMVT to an EP_XXXX function set. A discussion of the meaning of the table entries appears after the table.

Table 10-37. Sense Data Returned by PUMS After Receipt of an NMVT Request

Order of Checking	Mandatory / Optional	Sense Data	Condition	How Sent
1	Mandatory	X'1003 0001'	NMVTs not supported	-RSP
2	Optional	X'0815 0003'	NMVT being processed	-RSP
4	Optional	X'086F 0001'	MV length wrong for RU length	-RSP
5	Mandatory	X'080C 0005'	NMVT MV key not supported	-RSP

Note: This explanation applies to tables for the individual major vectors that follow, as well as to the previous table.

The column headed "Order of Checking" in each table indicates the order in which the different checks on a request are to be made. A check marked "Optional" in the second column does not have to be done at all by a product implementing PUMS, but if it is done, then it must occur at the specified place in the sequence of checks. Checks that have the same number in the "Order of Checking" column may be done in any order. Checks that have multiple entries in this column will be done at different times for different parts of an NMVT. A length error, for example, may be detected either when scanning for the command subvector or when scanning later for a required subvector.

Checks shown as mandatory must be performed by all implementations of PUMS that receive the specified NMVT request. Those marked as optional may or may not be performed by an implementation, but if they are performed the indicated sense data must be used to report that the check has failed.

The statements of the conditions reported by the different sense data are very abbreviated in the tables. *Systems Network Architecture Formats*, GA27-3136, should be consulted for the full definition of each sense data.

An entry of "-RSP or X'7D'" in the "How Sent" column indicates that different implementations of PUMS may elect to send the sense data either in a negative response or in a Sense Data (X'7D') subvector within a reply NMVT. For requests that specify multiple resources, however, these sense data may be sent in a -RSP only if they apply to *all* of the resources specified on the request. If the sense data applies to some, but not all, of the specified resources, it must be sent in a X'7D' subvector along with an SNA Address List (X'04') subvector that identifies the resources to which it applies.

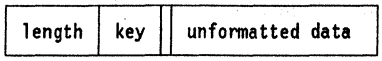
EP_XXXX Parsing of Individual Management Services Major Vectors

This section describes the general rules for parsing management services major vectors. The sections that follow it then list the individual parsing sequences for each of the management services major vectors that an EP_XXXX function set may receive. These parsing sequence begin after the common NMVT parsing, described in "Parsing of NMVTs" on page 10-152, has been completed.

Subvector Formats

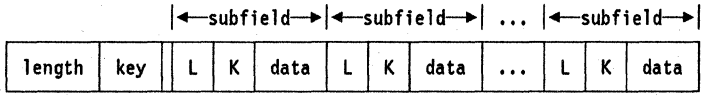
Management services subvectors fall into three categories, depending on the presence and placement of subfields within them. A *subfield*, like a subvector, consists of a length, a key, and data. Figure 10-65 on page 10-154 depicts the three types of subvectors defined.

Unformatted Subvector:



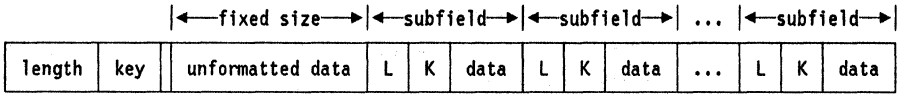
Example: SNA Address List (X'04') MS common subvector

Formatted Subvector:



Example: Failure Causes (X'96') Alert subvector

Partially Formatted Subvector:



Example: Product Identifier (X'11') MS common subvector

Key:
L: subfield length
K: subfield key

Figure 10-65. Unformatted, Formatted, and Partially Formatted Subvectors

An *unformatted* subvector contains no subfields at all. A *formatted* subvector contains only subfields after its own length and key. A *partially formatted* subvector contains a fixed number of bytes of unformatted data after its length and key, with the remainder consisting only of subfields.

There are a few general rules for the parsing of management services request major vectors.

1. Unrecognized subvectors must be skipped over in a request major vector, but unrecognized subfields within a formatted subvector must be reported. (See Rule 3 below for a clarification of what subvectors qualify as unrecognized.)

2. The command subvector must be identified first, then the remaining subvectors, if any.
3. The subvectors that may be recognized on a request are limited by the major vector and the command subvector.

Rule 3 requires further elaboration: After the receiver has completed the common parsing, and thereby determined that the request contains a management services major vector that it is prepared to support, it searches for the *command* subvector (Rule 2). A receiver has, for each request major vector, a list of command subvector keys that it supports for that major vector. If it cannot find one of these, it reports X'080C 0006' (Command Subvector Not Recognized).

If the receiver does find a command subvector, it examines its contents to determine what *other* subvectors are required for the execution of the command. The subvectors indicated, explicitly or implicitly, in the command subvector are not merely the set of subvectors that a receiver is required to process; they are also the *only* subvectors that a receiver is *permitted* to process. Rule 1 states that a receiver skips over unrecognized subvectors in a request major vector, and acts according to the remaining subvectors in the request that it does recognize. The set of additional subvectors that a receiver is *allowed* to recognize on a request, however, is limited by the command subvector. If, for example, the command subvector in a request indicates, explicitly or implicitly, that the X'04' subvector is not included, then a receiver *must* ignore a X'04' that was included erroneously, even though it can in general recognize and process a X'04' subvector.

To summarize, a receiver looks first for a command subvector, then for other subvectors indicated by the command subvector. Any other subvectors that may be present *must* be ignored.

Sense data X'080C 0006' (Command Subvector Not Recognized) is not, as it might appear, simply a variant of X'086C nn00' (Required Subvector Missing). If a request major vector supports several different command subvectors, any one of which is valid, then no *one* of them is individually required, and so sense data X'086C nn00' is not appropriate for reporting the case in which none of the possible command subvectors is recognized. (If this sense data were used, there would be no non-arbitrary way to select the key nn of "the" missing subvector.) Instead, sense data X'080C 0006' is used to report that no appropriate command subvector was recognized, even though the major vector key indicates that one must be present.

Parsing the Request RTM (X'8080') Major Vector

Table 10-38. Sense Data Returned by PUMS after Receipt of a Request RTM (X'8080') Major Vector

Order of Checking	Mandatory / Optional	Sense Data	Condition	How Sent
1	Mandatory	X'080C 0006'	Command subvector not recognized	-RSP
2	Mandatory[1]	X'086C nn00'	Required subvector missing	-RSP
2	Mandatory[2]	X'086C 0400'	X'04' subvector not first	-RSP
1,2	Optional	X'086F 0002'	Length error	-RSP
3	Mandatory[3]	X'1005 0001'	Wrong address type	-RSP
3	Mandatory[3]	X'080C 000A'	Multiple addresses not supported by receiver	-RSP
3	Mandatory	X'0870 nnxx'	Invalid value (unformatted sub-vector)	-RSP
1,2,3	Optional	X'086F nn05'	Length error	-RSP
4	Mandatory[3]	X'0806 0001'	Resources unknown	-RSP or X'7D'

Notes:

1. The only value allowed is nn = X'94', indicating the RTM Control subvector. This check is mandatory if the RTM Request (X'92') subvector indicates that the X'94' subvector is present.
2. This check is mandatory if the RTM Request (X'92') subvector indicates that the SNA Address List (X'04') subvector is present.
3. This check is mandatory if the SNA Address List (X'04') subvector is present.

Sense data X'0870 nn00' is sent for an invalid value in an unformatted sub-vector. For example, if the RTM measurement definition X'04'is not supported, sense data X'0870 9408' is used (byte 8 of X'94' subvector is in error).

Parsing the Request Product Set ID (X'8090') Major Vector

Table 10-39. Sense Data Returned by PUMS After Receipt of a Request Product Set ID (X'8090') Major Vector

Order of Checking	Mandatory / Optional	Sense Data	Condition	How Sent
1	Mandatory	X'080C 0006'	Command subvector not recognized	-RSP
1	Optional	X'086F 0002'	Length error	-RSP
1,2	Optional[1]	X'086F nn05'	Length error	-RSP

Notes:

1. The only values allowed are nn = X'81' and nn = X'83' indicating the command subvectors for the Request Product Set ID (X'8090') major vector.

Parsing the Change Management Major Vectors

Table 10-40 (Page 1 of 2). Sense Data Returned by MS After Receipt of a Request Change Control (X'8050') Major Vector

Order of Checking	Mandatory / Optional	Sense Data	Condition	Alert Sent
1	Optional	X'080C 000D'	Post-test not supported	No
1	Optional	X'080C 000E'	Prohibition of automatic removal not supported	No
1	Optional	X'084C 0008'	Volume not mounted	No
2	Mandatory	X'0838 0014'	Precluded combination of Install parameters	No
2	Mandatory	X'1005 0007'	SNA/MS command incompatible with SNA/FS server instruction	Yes
2	Optional	X'0838 0008'	Pre-test not applicable	No
2	Optional	X'0838 000A'	Automatic removal not applicable	No
2	Optional	X'0838 000B'	Post-test not applicable	No
3	Mandatory	X'0838 0001'	State exception	No
3	Mandatory	X'0838 0002'	State exception	No
3	Mandatory	X'0838 0003'	State exception	No
3	Mandatory	X'0838 0004'	State exception	No
3	Mandatory	X'0838 0005'	State exception	No
3	Mandatory	X'0838 0006'	State exception	No
3	Mandatory	X'0838 0007'	State exception	No
3	Mandatory	X'0838 000D'	State exception	No
3	Mandatory	X'0838 000E'	State exception	No
3	Mandatory	X'0838 0011'	State exception	No
3	Mandatory	X'0838 0012'	State exception	No
3	Mandatory	X'0838 0013'	State exception	No
3	Mandatory	X'0838 0015'	State exception	No
3	Mandatory	X'0838 0016'	State exception	No
3	Mandatory	X'081D 0001'	NETID.LUNAME in token string does not identify the receiver	No

Table 10-40 (Page 2 of 2). Sense Data Returned by MS After Receipt of a Request Change Control (X'8050') Major Vector				
Order of Checking	**Mandatory / Optional**	**Sense Data**	**Condition**	**Alert Sent**
3	Optional	X'0838 000F'	State exception	No
3	Optional	X'0838 0010'	State exception	No

Table 10-41. Sense Data Returned by MS After Receipt of a Request Activation (X'8066') Major Vector				
Order of Checking	**Mandatory / Optional**	**Sense Data**	**Condition**	**Alert Sent**
1	Mandatory	X'0872 0001'	Sessions are active	No
1	Optional	X'080C 000F'	Production-only activation not supported from focal point	No
1	Optional	X'084C 0008'	Volume not mounted	No

Parsing the Common Operations Services Major Vectors

Table 10-42. Sense Data Returned by PUMS after Receipt of a Common Operations Services Major Vector				
Order of Checking	**Mandatory / Optional**	**Sense Data**	**Condition**	**How Sent**
Sense Data Created by EP_COMMON_OPERATIONS_SERVICES*:*				
1	Mandatory	X'086C nn00'[1]	Required subvector missing	-RSP or X'7D'
2	Mandatory	X'086D nnmm'[2]	Required subfield missing	-RSP or X'7D'
3	Mandatory	X'8018 0001'	Specified application unknown	-RSP or X'7D'
Sense Data Created by a Served Network Management Application[3]*:*				
—	—	X'084B 0003'	Application not available	X'7D'
—	—	X'1003 000D'	Request not supported by application	X'7D'
—	—	X'081C 0n0m'[4]	Execution error	X'7D'
—	—	X'086F 0001'	Invalid major vector length	X'7D'
—	—	X'086D nnmm'[5]	Required subfield missing	X'7D'
—	—	X'080C 0006'	Command subvector not recognized	X'7D'
—	—	X'080C nn00'[6]	Required subvector missing	X'7D'
—	—	X'0806 0001'	Resource unknown	X'7D'
—	—	X'086A nnmm'[7]	Subfield key invalid	X'7D'
—	—	X'086B nnmm'[8]	Subfield value invalid	X'7D'
—	—	X'086F nn05'[9]	Subvector length error	X'7D'
—	—	X'086F nn06'[10]	Subfield length error	X'7D'

Notes:

1. The only value allowed is nn = X'06', indicating the Name List subvector.
2. The only values allowed are nn = X'06', mm= X'50', indicating the Destination Application Name subfield in the Name List subvector.
3. Neither an order of checking nor a set of mandatory checks is specified for served applications.
4. The digits n and m are used as follows:
   - n represents the link connection status:

     n = X'A': The link connection status has not changed from the state previous to the execution.

     n = X'B': The link connection status was modified from the state existing previous to the execution.

   - m identifies the nature of the execution error:

m = X'1': volatile storage error

m = X'2': nonvolatile storage (e.g., file access error)

m = X'3': link connection component (e.g., modem) interface error

m = X'4': unspecified software error.

5. The only values allowed are nn = X'06', mm= X'01', indicating the Resource Identifier subfield in the Name List subvector.
6. The only values allowed are nn = X'31' and nn = X'80', indicating, respectively, the Self-Defining Text Message subvector and the Test Setup Data subvector.
7. The key of the subvector containing the invalid subfield is identified in nn, and the key of the invalid subfield is identified in mm.
8. The key of the subvector containing the subfield with the invalid value is identified in nn, and the key of the subfield with the invalid value is identified in mm.
9. The key of the subvector with the invalid length is identified in nn.
10. The key of the subvector containing the subfield with the invalid length is identified in nn.

Part 3. Detailed Reference Material

Part III

Appendix A. Alerts Defined for Specific Environments

This appendix contains several groups of Alerts defined for specific environments. For each Alert, the following information is provided:

- The 32-bit Alert ID Number, calculated by means of the algorithm described in "Identification of Unique Alerts" on page 10-27
- The Alert Type
- The Alert Description
- The Probable Causes
- The User, Install, and Failure Causes (as appropriate), together with their associated qualifiers
- A partial list of the additional subvectors to be included in the Alert; required subvectors such as the Product Set ID are not listed

For certain Alerts, the Recommended Actions associated with the User, Install, and Failure Causes are also included. For other Alerts, the actions cannot be specified, since they are dependent on characteristics of the particular Alert sender, e.g., how the product is serviced, whether it is attended, or whether it is locally or remotely attached to the Alert receiver.

Alerts for Local Area Networks

Token-Ring LAN Alerts

This section documents the Alerts that should be sent by LAN managers and boundary-function-attached type 2.1 nodes. The following table defines which Alerts are sent by which nodes.

A 'Yes' in a column means that the Alert with this number is always sent by this type of node. A 'No' in a column means that the Alert with this number is never sent by this type of node.

Table A-1 (Page 1 of 2). Token-Ring Alert Sending Products

Alert Number	LAN Manager	BF-attached Type 2.1 Nodes
01	Note 1	Note 1
02	Note 1	Note 1
03	Note 1	Note 1
04	Note 1	Note 1
05	Note 1	Note 1
06	Yes	Yes

Table A-1 (Page 2 of 2). Token-Ring Alert Sending Products		
Alert Number	**LAN Manager**	**BF-attached Type 2.1 Nodes**
07	Yes	Yes
08	Yes	Yes
09	Note 3	Note 2
10	Note 3	No
11	No	Note 2
12	Note 3	Note 2
13	Note 3	Note 2

Notes:

1. This Alert flows if the sending product is unattended at the time of the error.
2. This Alert flows if a token-ring LAN manager is not present in the LAN to report errors on this ring.
3. If there are several token-ring LAN managers in a LAN, only one sends this Alert for each ring.

Token-Ring LAN Alert 1

Alert Condition:

The adapter detected a problem on its lobe during the wrap-test portion of the insertion process. The insertion process did not complete.

Alert ID Number		X'55BF3E1C'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open Failure
Probable Causes	X'3702'	Token-ring lobe
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3320' X'3711' X'3434'	Local token-ring adapter Local access unit Local lobe cables
Actions	X'1009' X'3301' X'2010' X'3101' X'32C0' X'82' SF X'82' SF	Attempt to re-open the adapter after 30 seconds If problem persists then do the following: Review link detailed data Contact token-ring administrator responsible for this LAN Report the following: (Adapter Number) (Error Code)

Additional SVs	X'51' SV X'03' SF X'23' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional)

Token-Ring LAN Alert 2

Alert Condition:

The adapter detected a beaconing condition on the ring during the insertion process. The insertion process did not complete.

Alert Identifier		X'CAF3C58A'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open Failure
Probable Causes	X'3703'	Token-ring fault domain
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3703'	Token-ring fault domain
Actions	X'1009' X'3301' X'2010' X'3101' X'32C0' X'82' SF X'82' SF	Attempt to re-open the adapter after 30 seconds If problem persists then do the following: Review link detailed data Contact token-ring administrator responsible for this LAN Report the following: (Adapter Number) (Error Code)
Additional SVs	X'51' SV X'06' SF X'26' SF X'07' SF X'05' SV X'10' SF	LAN LCS Data Token-Ring Fault Domain Description Fault Domain Names (Optional) Beacon Data Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh or 'UNKNOWN' Type=X'2E' (RING)

Token-Ring LAN Alert 3

Alert Condition:

The adapter detected the presence of a station with its individual address on the ring during the insertion process. The insertion process did not complete.

Alert ID Number		X'D615A61E'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open Failure
Probable Causes	X'3704'	Token-ring duplicate station address
User Causes	(none)	
Install Causes	X'3704'	Token-ring duplicate station addresses assigned
Actions	X'2010' X'3101' X'32C0' X'82'SF X'82'SF	Review link detailed data Contact token-ring administrator responsible for this LAN Report the following: (Adapter Number) (Error Code)
Failure Causes	(none)	
Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN)

Token-Ring LAN Alert 4

Alert Condition:

The adapter received a Remove Ring Station MAC frame during the insertion process. The insertion process did not complete.

Alert ID Number		X'44D1AD86'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open Failure
Probable Causes	X'3705'	Token-ring remove command received
User Causes	X'7101'	Token-ring remove adapter command received
Actions	X'2010' X'3101' X'32C0' X'82'SF X'82'SF	Review link detailed data Contact token-ring administrator responsible for this LAN Report the following (Adapter Number) (Error Code)
Install Causes	(none)	
Failure Causes	(none)	

Additional SVs	X'51' SV X'03' SF X'23' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional)

Token-Ring LAN Alert 5

Alert Condition:

An error was detected during the insertion process that is not defined in "Token-Ring LAN Alert 1," "Token-Ring LAN Alert 2," "Token-Ring LAN Alert 3," or "Token-Ring LAN Alert 4." These conditions are not expected to occur, so they are included within one Alert definition. The insertion process did not complete.

Alert ID Number		X'016E5F4E'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open failure
Probable Causes	X'3702' X'3701'	Token-ring lobe Token-ring LAN component
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3712' X'3701' X'2600'	Local token-ring lobe Token-ring LAN component Interference
Actions	X'2010' X'3101' X'32C0' X'82'SF X'82'SF	Review link detailed data Contact token-ring administrator responsible for this LAN Report the following: (Adapter Number) (Error Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh or 'UNKNOWN' Type=X'2E' (RING)

Token-Ring LAN Alert 6

Alert Condition:

The reporting station's adapter detected a wire-fault condition on the ring.

Alert ID Number		X'A676B230'
Alert Type	X'01'	Permanent
Alert Description	X'3212'	Wire fault
Probable Causes	X'3702'	Token-ring lobe
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3711' X'3434' X'3320'	Local access unit Local lobe cables Local token-ring adapter
Actions	X'2010' X'3101' X'0105' X'32C0' X'82' X'82'	Review link detailed data Contact token-ring administrator responsible for this LAN Request verification of management server reporting links [1] Report the following: (Adapter Number) (Error Code)
Additional SVS	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh (Hex ring ID) Type=X'2E' (RING)

Notes:

1. This code point is present if the sending product is a LAN manager and has reporting links with remote management servers.

Token-Ring LAN Alert 7

Alert Condition:

The reporting station's adapter has left the ring as part of the beacon automatic-recovery process. That is, the reporting station's adapter was a member of the beacon fault domain and removed itself from the ring to perform a self test, which was unsuccessful.

Alert ID Number		X'EB61E14F'
Alert Type	X'01'	Permanent
Alert Description	X'3213'	Auto-removal
Probable Causes	X'3702'	Token-ring lobe
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3320' X'3711' X'3434'	Local token-ring adapter Local access unit Local lobe cables
Actions	X'2010' X'3101' X'0105' X'32C0' X'82' X'82'	Review link detailed data Contact token-ring administrator responsible for this LAN Request verification of management server reporting links [1] Report the following: (Adapter Number) (Error Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh (Hex ring ID) Type=X'2E' (RING)

Notes:

1. This code point is present if the sending product is a LAN manager and has reporting links with remote management servers.

Token-Ring LAN Alert 8

Alert Condition:

The reporting station's adapter received a Remove Adapter command from a LAN manager and, as a result, left the LAN.

Alert ID Number		X'59F32622'
Alert Type	X'01'	Permanent
Alert Description	X'3214'	Remove adapter command received
Probable Causes	X'7013'	LAN manager operator
User Causes	X'7101'	Token-ring remove adapter command received

Actions	X'2010' X'3101' X'0105' X'32C0' X'82' X'82'	Review link detailed data Contact token-ring administrator responsible for this LAN Request verification of management server reporting links [1] Report the following: (Adapter Number) (Error Code)
Install Causes	(none)	
Failure Causes	(none)	
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional)

Notes:

1. This code point is present if the sending product is a LAN manager and has reporting links with remote management servers.

Token-Ring LAN Alert 9

Alert Condition:

The ring has been beaconing for a time longer than the hard-error detection timer. Manual intervention is necessary to recover the ring.

Alert ID Number		X'2102FCEB'
Alert Type	X'01'	Permanent
Alert Description	X'3215'	Token-ring inoperative
Probable Causes	X'3703'	Token-ring fault domain
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3703'	Token-ring fault domain
Actions	X'2010' X'3101' X'0105' X'32A0' X'82'SF	Review link detailed data Contact token-ring administrator responsible for this LAN Request verification of management server reporting links [1] Report the following: (Ring Status)

Additional SVs	X'51' SV X'06' SF X'26' SF X'07' SF X'05' SV X'10' SF	LAN LCS Data Token-Ring Fault Domain Description Fault Domain Names (Optional) Beacon Data Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh (Hex ring ID) Type=X'2E' (RING)

Notes:

1. This code point is present if the sending product is a LAN manager and has reporting links with remote management servers.

Token-Ring LAN Alert 10

Alert Condition:

The ring was in a beaconing condition for a time shorter than the hard-error detection timer. When the stations in the beacon fault domain were queried, one or both of them had left the ring.

Alert ID Number		X'698CCD51'
Alert Type	X'01'	Permanent
Alert Description	X'3213'	Auto removal
Probable Causes	X'3714'	Remote token-ring lobe
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3321' X'3713' X'3435'	Remote token-ring adapter Remote access unit Remote lobe cables
Actions	X'2010' X'3101' X'0105'	Review link detailed data Contact token-ring administrator responsible for this LAN Request verification of management server reporting links [1]

Additional SVs	X'51' SV X'06' SF X'26' SF X'08' SF X'28' SF X'05' SV X'10' SF	LAN LCS Data Fault Domain Description (CP) [2] Fault Domain Names (CP) [3] Single Individual MAC Address (CP) [4] Single Individual MAC Name (CP) [5] Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh (Hex ring ID) Type=X'2E' (RING)

Notes:

1. This code point is present if the sending product is a LAN manager and has reporting links with remote management servers.
2. This subfield is present if the sending product has determined that both beacon fault domain stations left the ring as part of the automatic recovery process.
3. This subfield is optionally present, but only present if the sending product has determined that both beacon fault domain stations left the ring as part of the automatic recovery process.
4. This subfield is present if the sending product has determined that only one of the beacon fault domain stations left the ring as part of the automatic recovery process.
5. This subfield is optionally present, but only present if the sending product has determined only one of the beacon fault domain stations left the ring as part of the automatic recovery process.

Token-Ring LAN Alert 11

Alert Condition:

The ring was in a beaconing condition for less than 52 seconds and then recovered. The sender of this Alert either knows that neither station in the fault domain left the ring, or has no knowledge about whether a station removed itself from the ring in order to bypass the fault.

Alert ID Number		X'2F36696E'
Alert Type	X'01'	Permanent
Alert Description	X'3216'	Token-ring temporary error
Probable Causes	X'3703'	Token-ring fault domain
User Causes	(none)	
Install Causes	(none)	

Failure Causes	X'3703'	Token-ring fault domain
Actions	X'2010' X'3101' X'0105'	Review link detailed data Contact token-ring administrator responsible for this LAN Request verification of management server reporting links [1]
Additional SVs	X'51' SV X'06' SF X'26' SF X'07' SF X'05' SV X'10' SF	LAN LCS Data Token-Ring Fault Domain Description Fault Domain Names (Optional) Beacon Data Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh (Hex ring ID) Type=X'2E' (RING)

Notes:

1. This code point is present if the sending product is a LAN manager and has reporting links with remote management servers.

Token-Ring LAN Alert 12

Alert Condition:

The ring error monitor (REM) has detected excessive soft errors for the ring.

Alert ID Number		X'A9998C16'
Alert Type	X'11'	Impending problem
Alert Description	X'4001'	Excessive token-ring errors
Probable Causes	X'3703'	Token-ring fault domain
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3703'	Token-ring fault domain
Actions	X'2010' X'3101'	Review link detailed data Contact token-ring administrator responsible for this LAN

Additional SVs	X'51' SV X'06' SF X'26' SF X'09' SF X'05' SV X'10' SF	LAN LCS Data Token-Ring Fault Domain Description Fault Domain Names (Optional) Fault Domain Error Weight Pair Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=RINGhhhh (Hex ring ID) Type=X'2E' (RING)

Token-Ring LAN Alert 13

Alert Condition:

The ring error monitor (REM) has detected that an adapter is experiencing excessive congestion and is discarding a significant number of frames.

Alert ID Number		X'57D16A21'
Alert Type	X'03'	Performance
Alert Description	X'5011'	Communication overrun
Probable Causes	X'3223'	Token-ring adapter interface
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1022' X'3324'	Communication program Token-ring adapter
Actions	X'2010' X'3101'	Review link detailed data Contact token-ring administrator responsible for this LAN
Additional SVs	X'51' SV X'02' SF X'08' SF X'28' SF X'05' SV X'10' SF	LAN LCS Data Ring or Bus Identifier Single Individual MAC Address Single Individual MAC Name Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN)

CSMA/CD LAN Alerts

This section documents the Alerts that should be sent by LAN managers and boundary-function attached T2.1 nodes for problems detected on CSMA/CD LANs. The following table defines which Alerts are sent by LAN managers and boundary-function attached T2.1 nodes.

A 'Yes' in a column means that the Alert with this number is always sent by this type of station. A 'No' in a column means that the Alert with this number is never sent by this type of station.

Table A-2. CSMA/CD LAN Alert Sending Products

Alert Number	LAN Manager	BF-Attached Type 2.1 Nodes
01	Note 1	Note 1
02	Note 1	Note 1
03	Note 1	Note 1
04	Note 1	Note 1
05	Yes	Yes
06	Note 3	Note 2
07	Yes	Yes
08	Yes	Yes
09	Yes	Yes
10	Yes	No
11	Yes	No

Notes:

1. This Alert flows if the sending product is unattended at the time of the error.
2. This Alert flows if a CSMA/CD LAN manager is not present in the LAN to report errors on this bus.
3. If there are several CSMA/CD LAN managers in a LAN, only one sends this Alert for each bus.

CSMA/CD LAN Alert 1

Alert Condition:

The adapter could not detect a carrier signal on the CSMA/CD network during the insertion process. The insertion process did not complete.

Alert ID Number		X'75CB6673'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open failure

Probable Causes	X'3436'	Local CSMA/CD LAN cable
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3436' X'3721'	Local CSMA/CD LAN cable CSMA/CD LAN component
Actions	X'1320' X'2010' X'3102' X'32A0' X'82'SF	Check cable connection and retry Review link detailed data Contact CSMA/CD administrator responsible for this LAN Report the following: (Adapter Number)
Additional SVs	X'51' SV X'02' SF X'05' SV X'10' SF	LAN LCS Data Ring/Bus Identifier Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) or 'UNKNOWN' Type=X'32' (CBUS)

CSMA/CD LAN Alert 2

Alert Condition:

The adapter detected the presence of a station with its address on the CSMA/CD LAN bus during the insertion process. The insertion process did not complete.

Alert ID Number		X'DD8A0144'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open failure
Probable Causes	X'3724'	CSMA/CD duplicate station address
User Causes	(none)	
Install Causes	X'3724'	Duplicate station address assigned
Actions	X'2010' X'3102' X'32A0' X'82'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN Report the following: (Adapter Number)
Failure Causes	(none)	

Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) or 'UNKNOWN' Type=X'32' (CBUS)

CSMA/CD LAN Alert 3

Alert Condition:

The adapter received a Remove Station command during the insertion process. The insertion process did not complete.

Alert ID Number		X'C6B5D6A5'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open failure
Probable Causes	X'3725'	CSMA/CD remove adapter command received
User Causes	X'7107'	CSMA/CD remove adapter command received
Actions	X'2010' X'3102' X'32A0' X'82'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN Report the following: (Adapter Number)
Install Causes	(none)	
Failure Causes	(none)	
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF	LAN LCS Data Ring/Bus Identifier [1] Local Individual MAC Address Local Individual MAC Name (Optional)

Notes:

1. This subfield is present only if the station has completed the portion of the open process during which it learns the ring identifier.

CSMA/CD LAN Alert 4

Alert Condition:

An error was detected during the insertion process that is not defined in "CSMA/CD LAN Alert 1" on page A-13, "CSMA/CD LAN Alert 2" on page A-14, or "CSMA/CD LAN Alert 3" on page A-15. The insertion process did not complete.

Alert ID Number		X'8B1836C5'
Alert Type	X'01'	Permanent
Alert Description	X'3211'	Open failure
Probable Causes	X'3721'	CSMA/CD LAN component
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3721'	CSMA/CD LAN component
Actions	X'1320' X'2010' X'3102' X'32C0' X'82' X'82'	Check cable connection and retry Review link detailed data Contact CSMA/CD administrator responsible for this LAN Report the following: (Adapter Number) (Error Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex Bus Name) or 'UNKNOWN' Type=X'32' (CBUS)

CSMA/CD LAN Alert 5

Alert Condition:

The reporting node's adapter received a Remove Adapter command from a LAN manager and, as a result, left the CSMA/CD LAN.

Alert ID Number		X'EB1D6ABB'
Alert Type	X'01'	Permanent
Alert Description	X'3214'	Remove adapter command received
Probable Causes	X'7013'	LAN manager operator

User Causes	X'7107'	CSMA/CD remove adapter command received
Actions	X'2010' X'3102' X'32A0' X'82'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN Report the following: (Adapter Number)
Install Causes	(none)	
Failure Causes	(none)	
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF	LAN LCS Data Ring/Bus Identifier Local Individual MAC Address Local Individual MAC Name (Optional)

CSMA/CD LAN Alert 6

Alert Condition:

A continuous-carrier condition has been detected on the CSMA/CD bus and the sending product's adapter is not the cause of the error.

Alert ID Number		X'43AAE16E'
Alert Type	X'01'	Permanent
Alert Description	X'3220'	CSMA/CD bus inoperative
Probable Causes	X'3330'	Adapter hardware
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3323'	Remote CSMA/CD adapter
Actions	X'2010' X'3102'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN
Additional SVs	X'51' SV X'02' SF X'08' SF X'28' SF X'05' SV X'10' SF	LAN LCS Data Ring/Bus Identifier Single MAC Address [1] Single MAC Name [2] Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) Type=X'32' (CBUS)

Notes:

1. This subfield is present if the sending product has isolated the continuous carrier to a single station address (not its own).
2. This subfield is optionally present, but only present if the sending product has isolated the continuous carrier to a single station address.

CSMA/CD LAN Alert 7

Alert Condition:

A no-carrier condition was detected on the CSMA/CD bus.

Alert ID Number		X'668E036D'
Alert Type	X'01'	Permanent
Alert Description	X'3221'	CSMA/CD LAN communications lost
Probable Causes	X'3426'	CSMA/CD LAN cables
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3436' X'3426' X'3721'	Local CSMA/CD adapter cable CSMA/CD LAN cables CSMA/CD LAN component
Actions	X'2010' X'3102' X'32A0' '82'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN Report the following: (Adapter Number)
Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) Type=X'32' (CBUS)

CSMA/CD LAN Alert 8

Alert Condition:

A continuous-carrier condition has been detected on the CSMA/CD bus and the source of the condition has been isolated to the local adapter.

Alert ID Number		X'176D9CDF'

Alert Type	X'01'	Permanent
Alert Description	X'3220'	CSMA/CD bus inoperative
Probable Causes	X'3322'	Local CSMA/CD adapter
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3322' X'3722'	Local CSMA/CD adapter CSMA/CD LAN translator unit
Actions	X'2010' X'3102'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN
Additional SVs	X'51' SV X'03' SF X'23' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) Type=X'32' (CBUS)

CSMA/CD LAN Alert 9

Alert Condition:

A continuous-carrier condition has been detected on the CSMA/CD bus and the source of the condition can not be isolated.

Alert ID Number		X'F7A377AE'
Alert Type	X'01'	Permanent
Alert Description	X'3220'	CSMA/CD bus inoperative
Probable Causes	X'3325'	CSMA/CD adapter
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3325' X'3722'	CSMA/CD adapter CSMA/CD LAN translator unit
Actions	X'2010' X'3102'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN

Additional SVs	X'51' SV X'02' SF X'05' SV X'10' SF	LAN LCS Data Ring or Bus Identifier Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) Type=X'32' (CBUS)

CMSA/CD LAN Alert 10

Alert Condition:

The LAN manager has detected that an adapter is experiencing excessive congestion and is discarding a significant number of frames.

Alert ID Number		X'1A56BAAE'
Alert Type	X'03'	Performance
Alert Description	X'5011'	Communication overrun
Probable Causes	X'3221'	CSMA/CD Adapter Interface
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1022' X'3325'	Communication program CSMA/CD adapter
Actions	X'2010' X'3101'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN
Additional SVs	X'51' SV X'02' SF X'08' SF X'28' SF X'05' SV X'10' SF	LAN LCS Data Ring or Bus Identifier Single Individual MAC Address Single Individual MAC Name Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN)

CSMA/CD LAN Alert 11

Alert Condition:

A continuous-carrier condition has been detected on the CSMA/CD bus and the source of the condition has been isolated to a remote adapter. The remote

adapter automatically removed itself from the LAN and the error has been bypassed.

Alert ID Number		X'C1A08052'
Alert Type	X'01'	Permanent
Alert Description	X'3213'	Auto removal
Probable Causes	X'3323'	Remote CSMA/CD adapter
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3325'	CSMA/CD adapter
Actions	X'2010' X'3102'	Review link detailed data Contact CSMA/CD administrator responsible for this LAN
Additional SVs	X'51' SV X'04' SF X'24' SF X'05' SV X'10' SF	LAN LCS Data Remote Individual MAC Address Remote Individual MAC Name Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=CBUShhhh (Hex bus number) Type=X'32' (CBUS)

Bridged LAN Alerts

This section defines the Alerts sent for problems associated with bridged LANs. These Alerts are sent only by LAN managers.

Bridged LAN Alert 1

Alert Condition:

An abnormally large percentage of frames are being discarded at a bridge. The bridge server has calculated the ratio of the number of frames it discarded to the number of frames it copied for forwarding and this ratio exceeded a threshold. Only the controlling LAN manager sends this Alert.

Alert ID Number		X'EE64FB52'
Alert Type	X'01'	Permanent
Alert Description	X'4010'	Error-to-traffic ratio exceeded
Probable Causes	X'3740'	LAN bridge
User Causes	X'7109' X'710A'	LAN bridge operator took bridge offline LAN manager operator took bridge offline

Install Causes	(none)	
Failure Causes	X'3700' X'3741' X'2007'	LAN component Congestion in LAN bridge LAN communications error
Actions	X'2010' X'3103'	Review link detailed data Contact LAN administrator responsible for this LAN
Additional SVs	X'51' SV X'0A' SF X'05' SV X'10' SF	LAN LCS Data Bridge Identifier Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=(bridge name) Type=X'3A' (BRDG)

Bridged LAN Alert 2

Alert Condition:

The LAN bridge was taken off-line by an operator. Shut-down was orderly and the LAN bridge server issued a frame warning the LAN managers that it was being removed from the LAN. Bridge frame forwarding functions were terminated. That is, either an operator at the bridge or at a LAN manager issued a Set Bridge Parameters frame to set the Route Active parameter of the bridge server to cause the bridge not to forward frames. Only the controlling LAN manager sends this Alert.

Alert ID Number		X'608A29AF'
Alert Type	X'01'	Permanent
Alert Description	X'B003'	LAN bridge taken offline
Probable Causes	X'7012' X'7013'	LAN bridge operator LAN manager operator
User Causes	X'7109' X'710A'	LAN bridge operator took bridge offline LAN manager operator took bridge offline
Actions	X'2010' X'3103'	Review link detailed data Contact LAN administrator responsible for this LAN
Install Causes	(none)	
Failure Causes	(none)	

Additional SVs	X'51' SV X'0A' SF X'05' SV X'10' SF	LAN LCS Data Bridge Identifier Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender: Name=(bridge name) Type=X'3A' (BRDG)

Bridged LAN Alert 3

Alert Condition:

A LAN reporting link to remote management servers has been lost. The remote link station does not respond. The inactivity timer (Ti) has expired, causing the remote station to be polled. The remote station does not respond to the poll.

The controlling LAN manager sends this Alert when it loses reporting links and it's local ring was not in a beaconing condition for a pre-determined period before the link was lost. The default for this pre-determined period is 1 minute, but it is configurable. Only the controlling LAN manager sends this Alert.

Alert ID Number		X'841206FE'
Alert Type	X'01'	Permanent
Alert Description	X'3330'	Management server reporting link error
Probable Causes	X'2107'	LAN LLC communications/remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2107' X'F017'	LAN LLC communications/remote node Poll count exhausted
Actions	X'2010' X'3103' X'32C0' X'82' SF X'82' SF	Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Name=(LAN name)
		Type=X'39' (LAN)
		Second resource below sender:[2]
		Name=(bridge name)
		Type=X'3A' (BRDG)
		Second resource below sender:[3]
		Name=(management server name)
		Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 4

Alert Condition:

A LAN reporting link to remote management servers has been lost. The remote link station sent a Disconnect Mode to the local link station. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful. Only the controlling LAN manager sends this Alert.

Alert ID Number		X'B1D9A4C5'
Alert Type	X'01'	Permanent

Alert Description	X'3300'	Management server reporting link error
Probable Causes	X'2007'	LAN LLC communications
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2007' X'F01A'	LAN LLC communications DM received
Actions	X'2010' X'3103' X'32C0' X'82' SF X'82' SF	Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender:[2] Name=(bridge name) Type=X'3A' (BRDG) Second resource below sender:[3] Name=(management server name) Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 5

Alert Condition:

A LAN reporting link to remote management servers has been lost. The local link station sent an invalid or unsupported frame to the remote link station. This resulted in the remote link station returning a Frame Reject. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'8A5B2D2C'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F010'	Software program Frame reject received: Invalid/unsupported command or response sent
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional svs	X'51' SV	LAN LCS Data
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Name=(LAN name)
		Type=X'39' (LAN)
		Second resource below sender:[2]
		Name=(bridge name)
		Type=X'3A' (BRDG)
		Second resource below sender:[3]
		Name=(management server name)
		Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 6

Alert Condition:

A LAN reporting link to remote management servers has been lost. The local link station sent an I-frame when not permitted to the remote link station. This resulted in the remote link station returning a Frame Reject. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'8E9A309B'

Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F011'	Software program Frame reject received: I-frame sent when not permitted
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender:[2] Name=(bridge name) Type=X'3A' (BRDG) Second resource below sender:[3] Name=(management server name) Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 7

Alert Condition:

A LAN reporting link to remote management servers has been lost. The local link station sent a frame with an invalid N(r). This resulted in the remote link station returning a Frame Reject. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'83D91642'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F012'	Software program Frame reject received: invalid N(r) sent
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Name=(LAN name)
		Type=X'39' (LAN)
		Second resource below sender:[2]
		Name=(bridge name)
		Type=X'3A' (BRDG)
		Second resource below sender:[3]
		Name=(management server name)
		Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 8

Alert Condition:

A LAN reporting link to remote management servers has been lost. The local link station sent a frame with an I-field that was too long. This resulted in the remote link station returning a Frame Reject.

Alert ID Number		X'87180BF5'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error

Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F013'	Software program Frame reject received: Maximum I-field length exceeded
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender:[2] Name=(bridge name) Type=X'3A' (BRDG) Second resource below sender:[3] Name=(management server name) Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 9

Alert Condition:

A LAN reporting link to remote management servers has been lost. The remote link station sent an invalid or unsupported frame to the local link station. This resulted in the local link station returning a Frame Reject. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'28EF2B5D'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1023' X'F020'	Communications program in remote node Invalid/unsupported command or response received
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Name=(LAN name)
		Type=X'39' (LAN)
		Second resource below sender:[2]
		Name=(bridge name)
		Type=X'3A' (BRDG)
		Second resource below sender:[3]
		Name=(management server name)
		Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 10

Alert Condition:

A LAN reporting link to remote management servers has been lost. The remote link station sent an I-frame when not permitted to the local link station. This resulted in the local link station returning a Frame Reject. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'2C2E36EA'

Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes		(none)
Install Causes		(none)
Failure Causes	X'1023' X'F021'	Communications program in remote node I-Frame received when not permitted
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender:[2] Name=(bridge name) Type=X'3A' (BRDG) Second resource below sender:[3] Name=(management server name) Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 11

Alert Condition:

A LAN reporting link to remote management servers has been lost. The remote link station sent a frame with an invalid N(r). This resulted in the local link station returning a Frame Reject. The LAN manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'216D1033'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes		(none)
Install Causes		(none)
Failure Causes	X'1023' X'F022'	Communications program in remote node Invalid N(r) received
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Name=(LAN name)
		Type=X'39' (LAN)
		Second resource below sender:[2]
		Name=(bridge name)
		Type=X'3A' (BRDG)
		Second resource below sender:[3]
		Name=(management server name)
		Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 12

Alert Condition:

A LAN reporting link to remote management servers has been lost. The remote link station sent a frame with an I-field that was too long. This resulted in the local link station returning a Frame Reject. The LAN. manager tried to re-establish the link after a pre-determined time and the attempt was unsuccessful.

Alert ID Number		X'25AC0D84'

Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes		(none)
Install Causes		(none)
Failure Causes	X'1023' X'F023'	Communications program in remote node Received I-field exceeded maximum length
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' (LAN) Second resource below sender:[2] Name=(bridge name) Type=X'3A' (BRDG) Second resource below sender:[3] Name=(management server name) Type=X'3C' (MSVR)

Notes:

1. This subfield is present if the lost link traversed a LAN bridge.
2. This subfield is present if the management servers with which the link was lost were located in a LAN bridge.
3. This subfield is present if the management servers with which the link was lost were not located in a LAN bridge.

Bridged LAN Alert 13

Alert Condition:

The remote management server has been unable to receive data on the link and has been sending Receiver Not Ready frames contiguously for more than 30 seconds.

Alert ID Number		X'6CEB625D'
Alert Type	X'01'	Permanent
Alert Description	X'5004'	Out of resources
Probable Causes	X'1023' X'1031'	Communications program in remote node LAN management server
User Causes	X'710B'	User incapacitated LAN management server program
Actions	X'1405'	Reactivate LAN management server program
Install Causes	(none)	
Failure Causes	X'0020' X'0111' X'1031'	Excessive load on processor Number of LAN mgmt. frames received exceeds buffer capacity LAN management server
Actions	X'2010' X'3103'	Review link detailed data Contact LAN administrator responsible for this LAN
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] Hierarchy/Resource List (CP) Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' LAN Second resource below sender:[2] Name=(bridge name) Type=X'3A' BRDG Second resource below sender:[3] Name=(management server name) Type=X'3C' MSVR

Notes:

1. This subfield is present if the link about which this Alert pertains traversed a LAN bridge.
2. This subfield is present if the management servers with which the link identified in this Alert were located in a LAN bridge.
3. This subfield is present if the management servers with which the link identified in this Alert were not located in a LAN bridge.

Bridged LAN Alert 14

Alert Condition:

The remote management server has been congested and, as a result, has probably discarded management data.

Alert ID Number		X'68882FCE'
Alert Type	X'01'	Permanent
Alert Description	X'3230'	LAN management data lost
Probable Causes	X'1031' X'1023'	LAN management server Communications program in remote node
User Causes	X'710B'	User incapacitated LAN management server program
Actions	X'1405'	Reactivate LAN management server program
Install Causes	(none)	
Failure Causes	X'0020' X'0111' X'1031'	Excessive load on processor Number of LAN mgmt. frames received exceeds buffer capacity LAN management server
Actions	X'2010' X'3103'	Review link detailed data Contact LAN administrator responsible for this LAN
Additional SVs	X'51' SV X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'05' SV X'10' SF	LAN LCS Data Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] Hierarchy/Resource List (CP) Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' LAN Second resource below sender:[2] Name=(bridge name) Type=X'3A' BRDG Second resource below sender:[3] Name=(management server name) Type=X'3C' MSVR

Notes:

1. This subfield is present if the link about which this Alert pertains traversed a LAN bridge.
2. This subfield is present if the management servers with which the link identified in this Alert were located in a LAN bridge.
3. This subfield is present if the management servers with which the link identified in this Alert were not located in a LAN bridge.

Bridged LAN Alert 15

Alert Condition:

A LAN manager attempted to establish a reporting link with management servers, but was rejected because it used an invalid password. Only the controlling LAN manager sends this Alert.

Alert ID Number		X'91C0A9A7'
Alert Type	X'12'	Unknown
Alert Description	X'C001'	Invalid reporting link password
Probable Causes	X'7013'	Remote LAN manager operator
User Causes	X'7103' X'7104'	LAN manager operator entered incorrect password Unauthorized access to LAN management server attempted
Actions	X'3301' X'2010' X'3103'	If problem persists then do the following: Review link detailed data Contact LAN administrator responsible for this LAN
Install Causes	(none)	
Failure Causes	(none)	
Additional SVs	X'51' SV X'08' SF X'28' SF X'05' SV X'10' SF	LAN LCS Data Single MAC Address Single MAC Name (Optional) Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(LAN name) Type=X'39' LAN

SDLC/LAN LLC Alerts

LAN LLC Alerts

LAN LLC Alert 1

Alert Condition:

A LAN logical link has been lost. The remote link station does not respond. The inactivity timer (Ti) or acknowledgement timer (T1) has expired, causing the remote station to be polled. The remote station does not respond to the poll.

Alert ID Number		X'5B8F5BA7'
Alert Type	X'01'	Permanent
Alert Description	X'3300'	Link error
Probable Causes	X'2107'	LAN LLC communications/remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2107' X'F017'	LAN LLC communications/remote node Poll count exhausted
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'02' SF	Ring/Segment Identifier
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS Configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Resource Name=(LAN name)
		Resource Type=X'39' (LAN)
		Second resource below sender:
		Resource Name=(CP name)
		Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 2

Alert Condition:

A LAN logical link has been lost. The remote link station sent a Disconnect Mode response to the local link station.

Alert ID Number		X'B1D9A4C5'
Alert Type	X'01'	Permanent
Alert Description	X'3300'	Link error
Probable Causes	X'2007'	LAN LLC communications
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2007' X'F01A'	LAN LLC communications DM received

Actions	X'3303' X'2010' X'3101' X'32C0' X'82' SF X'82' SF	If problem persists then do the following Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 3

Alert Condition:

A LAN logical link has been lost. The remote link station sent a SABME command to the local link station which was already open (had been initialized via a SABME-UA exchange).

Alert ID Number		X'E65B0B7F'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error

Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1023' X'F016'	Communications program in remote node SABME received while in ABME
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 4

Alert Condition:

A LAN logical link has been lost. The local link station sent an invalid or unsupported command or response to the remote link station. This resulted in the remote link station returning a Frame Reject response.

Alert ID Number		X'8A5B2D2C'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F010'	Software program Frame reject received: Invalid/unsupported command or response sent
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'02' SF	Ring/Segment Identifier
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS Configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Resource Name=(LAN name)
		Resource Type=X'39' (LAN)
		Second resource below sender:
		Resource Name=(CP name)
		Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 5

Alert Condition:

A LAN logical link has been lost. The local link station sent an I-field when not permitted to the remote link station. This resulted in the remote link station returning a Frame Reject response.

Alert ID Number		X'8E9A309B'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	

Failure Causes	X'1000' X'F011'	Software program Frame reject received: I-field sent when not permitted
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 6

Alert Condition:

A LAN logical link has been lost. The local link station sent a frame with an invalid N(r). This resulted in the remote link station returning a Frame Reject response.

Alert ID Number		X'83D91642'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F012'	Software program Frame reject received: invalid N(r) sent
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 7

Alert Condition:

A LAN logical link has been lost. The local link station sent a frame with an I-field that was too long. This resulted in the remote link station returning a Frame Reject response.

Alert ID Number		X'87180BF5'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1000'	LAN LLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F013'	Software program Frame reject received: maximum I-field length exceeded
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'02' SF	Ring/Segment Identifier
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS Configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Resource Name=(LAN name)
		Resource Type=X'39' (LAN)
		Second resource below sender:
		Resource Name=(CP name)
		Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 8

Alert Condition:

A LAN logical link has been lost. The remote link station sent an invalid or unsupported command or response to the local link station. This resulted in the local link station returning a Frame Reject response.

Alert ID Number		X'28EF2B5D'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	

Failure Causes	X'1023' X'F020'	Communications program in remote node Invalid/unsupported command or response received
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 9

Alert Condition:

A LAN logical link has been lost. The remote link station sent an I-field when not permitted to the local link station. This resulted in the local link station returning a Frame Reject response.

Alert ID Number		X'2C2E36EA'

Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes		(none)
Install Causes		(none)
Failure Causes	X'1023' X'F021'	Communications program in remote node I-field received when not permitted
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 10

Alert Condition:

A LAN logical link has been lost. The remote link station sent a frame with an invalid N(r). This resulted in the local link station returning a Frame Reject response.

Alert ID Number		X'216D1033'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1023' X'F022'	Communications program in remote node Invalid N(r) received
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)

Additional SVs	X'51' SV	LAN LCS Data
	X'02' SF	Ring/Segment Identifier
	X'03' SF	Local Individual MAC Address
	X'23' SF	Local Individual MAC Name (Optional)
	X'04' SF	Remote Individual MAC Address
	X'24' SF	Remote Individual MAC Name (Optional)
	X'05' SF	LAN Routing Information [1]
	X'52' SV	LCS Configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Resource Name=(LAN name)
		Resource Type=X'39' (LAN)
		Second resource below sender:
		Resource Name=(CP name)
		Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

LAN LLC Alert 11

Alert Condition:

A LAN logical link has been lost. The remote link station sent a frame with an I-field that was too long. This resulted in the local link station returning a Frame Reject response.

Alert ID Number		X'25AC0D84'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2007' X'1023'	LAN LLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	

Failure Causes	X'1023' X'F023'	Communications program in remote node Received I-field exceeded maximum length
Actions	X'3301' X'2010' X'3103' X'32C0' X'82' SF X'82' SF	If problem persists then do the following: Review link detail data Contact LAN administrator responsible for this LAN Report the following (Adapter Number) (Reference Code)
Additional SVs	X'51' SV X'02' SF X'03' SF X'23' SF X'04' SF X'24' SF X'05' SF X'52' SV X'02' SF X'04' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LAN LCS Data Ring/Segment Identifier Local Individual MAC Address Local Individual MAC Name (Optional) Remote Individual MAC Address Remote Individual MAC Name (Optional) LAN Routing Information [1] LCS Configuration Remote Device Address Local Device Address Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(LAN name) Resource Type=X'39' (LAN) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

Notes:

1. This subfield is present if the lost logical link traversed a MAC bridge.

SDLC Alerts

SDLC Alert 1

Alert Condition:

An SDLC logical link has been lost. The secondary link station does not respond to poll frames sent by the primary station.

Alert ID Number		X'32A37F1B'
Alert Type	X'01'	Permanent
Alert Description	X'3300'	Link error
Probable Causes	X'2104' X'2031'	SDLC communications/remote node Line
User Causes	X'0209'	Remote device power off
Actions	X'0200'	Check power
Install Causes	(none)	
Failure Causes	X'2104' X'3511' X'F017'	SDLC communications/remote node Line Poll count exhausted
Actions		(Sender-specific actions)
Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 2

Alert Condition:

An SDLC logical link has been lost. The secondary link station sent a Disconnect Mode response to the primary link station.

Alert ID Number		X'BD84C4C9'
Alert Type	X'01'	Permanent

Alert Description	X'3300'	Link error
Probable Causes	X'2004'	SDLC communications
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2004' X'F01A'	SDLC communications DM received
Actions		(Sender-specific actions)
Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 3

Alert Condition:

An SDLC logical link has been lost. The primary link station sent a SNRM command to the secondary link station while it was in NRM.

Alert ID Number		X'D635CA1E'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1023'	SDLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	

Failure Causes	X'1023' X'F015'	Communications program in remote node SNRM received while in NRM
Actions		(Sender-specific actions)
Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 4

Alert Condition:

An SDLC logical link has been lost. The primary link station sent an invalid or unsupported command to the secondary link station. This resulted in the secondary link station returning a Frame reject response.

Alert ID Number		X'B776CA94'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1000'	SDLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F010'	Software program Frame reject received: invalid/unsupported command or response sent
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 5

Alert Condition:

An SDLC logical link has been lost. The primary link station sent an I-field when not permitted to the secondary link station. This resulted in the secondary link station returning a Frame reject response.

Alert ID Number		X'B3B7D723'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1000'	SDLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F011'	Software program Frame reject received: I-field sent when not permitted
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 6

Alert Condition:

An SDLC logical link has been lost. The primary link station sent a frame with an invalid N(r). This resulted in the secondary link station returning a Frame reject response.

Alert ID Number		X'BEF4F1FA'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1000'	SDLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F012'	Software program Frame reject received: invalid N(r) sent
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 7

Alert Condition:

An SDLC logical link has been lost. The primary link station sent a frame with an I-field that was too long. This resulted in the secondary link station returning a Frame reject response.

Alert ID Number		X'BA35EC4D'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1000'	SDLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F013'	Software program Frame reject received: maximum I-field length exceeded
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 8

Alert Condition:

An SDLC logical link has been lost. The secondary link station sent an invalid or unsupported command to the primary link station. This resulted in the primary link station returning a Frame reject response.

Alert ID Number		X'15C2CCE5'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1023'	SDLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1023' X'F020'	Communications program in remote node Invalid/unsupported command or response received
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 9

Alert Condition:

An SDLC logical link has been lost. The secondary link station sent an I-field when not permitted to the primary link station. This resulted in the primary link station returning a Frame reject response.

Alert ID Number		X'1103D152'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1023'	SDLC communications Communications program in remote node
User Causes		(none)
Install Causes		(none)
Failure Causes	X'1023' X'F021'	Communications program in remote node I-field received when not permitted
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 10

Alert Condition:

An SDLC logical link has been lost. The primary link station sent a frame with an invalid N(r). This resulted in the secondary link station returning a Frame reject response.

Alert ID Number		X'1C40F78B'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1023'	SDLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1023' X'F022'	Communications program in remote node Invalid N(r) received
Actions		(Sender-specific actions)

Additional SVs	X'52' SV	LCS Configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'06' SF	Link Station Attributes
	X'07' SF	Link Attributes
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Resource Name=(Adapter number)
		Resource Type=X'21' (Adapter)
		Second resource below sender:
		Resource Name=(CP name)
		Resource Type=X'F4' (CP)

SDLC Alert 11

Alert Condition:

An SDLC logical link has been lost. The number of I-frames transmitted by the remote link station has exceeded the local link station's receive window size.

Alert ID Number		X'EABB6A14'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1023'	SDLC communications Communications program in remote node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1023' X'F01B'	Communications program in remote node Receive window size exceeded
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 12

Alert Condition:

An SDLC logical link has been lost. The secondary link station's inactivity timer has expired.

Alert ID Number		X'0E2DDF11'
Alert Type	X'01'	Permanent
Alert Description	X'3300'	Link error
Probable Causes	X'2104' X'2031'	SDLC communications/remote node Line
User Causes	X'0209'	Remote device power off
Actions	X'0200'	Check power
Install Causes	(none)	
Failure Causes	X'2104' X'3511' X'F019'	SDLC communications/remote node Line Inactivity timer expired
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'8C' SV X'01' SF X'02' SF X'03' SF X'04' SF X'05' SF X'06' SF X'07' SF X'08' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Link Station Data Current Ns/Nr Counts Outstanding Frame Count Last Control Field Received Last Control Field Sent Sequence Number Modulus Link Station State LLC Reply Timer Expiration Count Last Received Nr Count Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 13

Alert Condition:

Link establishment has failed. The local link station's retry limit for XID has been exceeded.

Alert ID Number		'0AECC2A6'
Alert Type	X'01'	Permanent
Alert Description	X'3300'	Link error
Probable Causes	X'2104' X'2031'	SDLC communications/remote node Line
User Causes	X'0209'	Remote device power off
Actions	X'0200'	Check power
Install Causes	(none)	
Failure Causes	X'2104' X'3511' X'F018'	SDLC communications/remote node Line XID poll count exhausted
Actions		(Sender-specific actions)

Additional SVs	X'52' SV X'02' SF X'04' SF X'06' SF X'07' SF X'05' SV X'10' SF	LCS Configuration Remote Device Address Local Device Address Link Station Attributes Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Resource Name=(Adapter number) Resource Type=X'21' (Adapter) Second resource below sender: Resource Name=(CP name) Resource Type=X'F4' (CP)

SDLC Alert 14

Alert Condition:

An SDLC logical link has been lost. The secondary link station send a Frame reject response, but it contained no I-field indicating the reason for the rejection.

Alert ID Number		X'A472BC48'
Alert Type	X'01'	Permanent
Alert Description	X'2100'	Software program error
Probable Causes	X'2004' X'1000'	SDLC communications Software program
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'1000' X'F014'	Software program Frame reject received: no reason specified
Actions		(Sender-specific actions)

Additional SVs	X'52' SV	LCS Configuration
	X'02' SF	Remote Device Address
	X'04' SF	Local Device Address
	X'06' SF	Link Station Attributes
	X'07' SF	Link Attributes
	X'8C' SV	Link Station Data
	X'01' SF	Current Ns/Nr Counts
	X'02' SF	Outstanding Frame Count
	X'03' SF	Last Control Field Received
	X'04' SF	Last Control Field Sent
	X'05' SF	Sequence Number Modulus
	X'06' SF	Link Station State
	X'07' SF	LLC Reply Timer Expiration Count
	X'08' SF	Last Received Nr Count
	X'05' SV	Hierarchy/Resource List
	X'10' SF	Hierarchy Name List
		First resource below sender:
		Resource Name=(Adapter number)
		Resource Type=X'21' (Adapter)
		Second resource below sender:
		Resource Name=(CP name)
		Resource Type=X'F4' (CP)

Alerts for Switched Link Connections

X.21 and X.21 Short Hold Mode Alerts

This section documents the Alerts sent by nodes using X.21 and X.21 Short Hold Mode. These Alerts are sent by type 2 or boundary-function-attached type 2.1 nodes. These stations do not send Alerts for errors detected on the initial call, but only send Alerts for errors detected on reconnections. These Alerts are sent as delayed Alerts, and are sent at all only if the same type 4 node to which the reconnection attempt was directed is eventually reached.

X.21 Alert 1

Alert Condition:

The secondary station received a CPS01, but the X.21 network did not signal ready for data within a specified period of time (See GA27-3287 *IBM Implementation of X.21 Interface General Information Manual* for the value of the timer.) This indicates the outgoing call was signalled to the DTE, but no further response was received. This is sent as a delayed Alert by the secondary station.

Alert ID Number		X'861061E5'
Alert Type	X'01'	Permanent (Delayed)

Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2201'	Called DTE
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2201' X'2300'	Called DTE Connection not established
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 2

Alert Condition:

The secondary station received a CPS02, CPS03 or CPS05, but the X.21 network did not signal ready for data within a specified period of time (See GA27-3287 *IBM Implementation of X.21 Interface General Information Manual* for the value of the timer.) This indicates the outgoing call was received by the DTE, but no further response was received. This is sent as a delayed Alert by the secondary station.

Alert ID Number		X'D974044D'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2005' X'2300'	X.21 network Connection not established
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 3

Alert Condition:

The secondary station received a CPS04, but the X.21 network did not signal ready for data within a specified period of time (See GA27-3287 *IBM Implementation of X.21 Interface General Information Manual* for the value of the timer.) This indicates the outgoing call reached a private network, but no further response was received. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'C05198EA'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2401'	Private network reached
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2041' X'2300'	Private network reached Connection not established
Actions	X'3302' X'3104' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact network information service for private network called Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 4

Alert Condition:

Retries have been exhausted and a CPS20 was received by the secondary station indicating the called number could not be connected. No cause was specified. This is sent as a delayed Alert by the secondary station.

Alert ID Number		X'B511541E'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'FE00'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2005'	X.21 network

Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 5

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS21 was received by the secondary station indicating the called number was busy. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'F230538E'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2301'	Called number busy
User Causes	X'2301'	Called number busy
Actions	X'3302' X'3122' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact called DTE's operator Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)
Install Causes	(none)	
Failure Causes	(none)	

X.21 Alert 6

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS22 was received by the secondary station indicating there was a procedure error in the selection signals. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'C2E9214F'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'3309'	Line adapter

User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3309'	Line adapter
Actions	X'3302' X'30E1' X'83' SF X'3503' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact the service representative for (Product Identifier) Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 7

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS23 was received by the secondary station indicating there was a transmission error in the selection signals. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'9B1BDB20'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2036'	DCE-DSE connection
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3512'	The connection between the calling DCE and its DSE
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 8

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS41 was received by the secondary station. This indicates the calling DTE is not permitted connection to the called DTE. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'033AC8C8'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2308'	Access barred
User Causes	X'2304'	Incorrect telephone number specified
Actions	X'0103'	Verify telephone number
Install Causes	(none)	
Failure Causes	X'2308'	Access barred
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 9

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS42 was received by the secondary station. This indicates the called DTE has been assigned a new number. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'F0FE2CBD'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2306'	Changed Number
User Causes	X'2304'	Incorrect telephone number specified
Actions	X'0103'	Verify telephone number
Install Causes	(none)	
Failure Causes	X'2306'	New Telephone Number Assigned to Called DTE
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 10

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS43 was received by the secondary station. This indicates the called DTE address is out of the numbering plan or not assigned to any DTE. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'E8561A00'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2300'	Connection not established
User Causes	X'2304'	Incorrect telephone number specified
Actions	X'0103'	Verify telephone number
Install Causes	(none)	
Failure Causes	X'2307'	Called number outside numbering plan or unknown by network
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 11

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS44 was received by the secondary station. This indicates the called number is out of order. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'987DFF4D'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2201' X'3510' X'2005'	Called DTE Called DCE X.21 network
User Causes	X'0212' X'2510' X'0211'	Called DTE power off Line not enabled at called DTE Called DCE power off
Actions	X'1200' X'0200' X'1331'	Retry Check power Enable line then retry

Install Causes	(none)	
Failure Causes	X'2203' X'3513'	Called DTE signalling uncontrolled not ready Local loop associated with the called DTE
Actions	X'3302' X'3122' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact called DTE's operator Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 12

Alert Condition:

The retries have been exhausted when trying to connect or reconnect a link and a CPS45 was received by the secondary station. This indicates the called DTE is signalling controlled not ready. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'9E6C217A'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2201'	Called DTE
User Causes	X'2511'	Port deactivated at called DTE
Actions	X'1330'	Activate port then retry
Install Causes	(none)	
Failure Causes	X'2202'	Called DTE signalling controlled not ready
Actions	X'3302' X'3122' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact called DTE's operator Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 13

Alert Condition:

A CPS45 with a date and time was received by the secondary station. This indicates the called DTE is signalling controlled not ready. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'FEBADD77'

Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2201'	Called DTE
User Causes	X'2201'	Called DTE taken out of service
Actions	X'12C0' X'82' SF X'82' SF	Retry after (Year/Month/Day) (Time)
Install Causes	(none)	
Failure Causes	X'2202'	Called DTE signalling controlled not ready
Actions	X'3301' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem persists then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 14

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS46 was received by the secondary station. This indicates the called DTE is signalling uncontrolled not ready. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'7B66A32A'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2201'	Called DTE
User Causes	X'0212' X'2510'	Called DTE Line not enabled at called DTE
Actions	X'0200' X'1331'	Check power Enable line then retry
Install Causes	(none)	
Failure Causes	X'2203'	Called DTE signalling uncontrolled not ready
Actions	X'3302' X'3122' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact called DTE's operator Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 15

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS47 was received by the secondary station. This indicates the called DCE does not have power. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'ECECAA6D'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'3510'	Called DCE
User Causes	X'0211'	Called DCE power off
Actions	X'0200'	Check power
Install Causes	(none)	
Failure Causes	X'3510'	Called DCE
Actions	X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 16

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS48 was received by the secondary station. This indicates the requested facility is detected as invalid by the DCE at the local DTE/DCE interface. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'DF55F53F'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2307'	Invalid request
User Causes	X'2300'	Calling DTE does not subscribe to this facility
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number)

Install Causes	(none)	
Failure Causes	(none)	

X.21 Alert 17

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS49 was received by the primary station. This indicates an error with the local loop associated with the called DTE. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'A5D509ED'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3513'	Local loop associated with the called DTE
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 18

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS51 was received by the primary station. This indicates the called number was busy and the network information service should be contacted for details. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'F230538E'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2301'	Called number busy
User Causes	X'2301'	Called number busy

Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)
Install Causes	(none)	
Failure Causes	(none)	

X.21 Alert 19

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS52 was received by the primary station. This indicates the called DTE belongs to a user class of service which is incompatible with that of the calling DTE. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'367EF253'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2304'	Incorrect number called
User Causes	X'2304'	Incorrect telephone number specified
Actions	X'0103'	Verify telephone number
Install Causes	(none)	
Failure Causes	X'2309' X'230A'	Speed classes incompatible User classes of service incompatible
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 20

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS61 was received by the station. This indicates the network is temporarily congested. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'D55C05B3'

Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'5001'	Network congestion
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3521'	Temporary lack of resources in X.21 network
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 21

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS71 was received by the station. This indicates a shortage of network resources. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'D81F236A'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'5001'	Network congestion
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3522'	Long term lack of resources in X.21 network
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 22

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and a CPS72 was received by the station. This indicates the RPOA nominated by the calling DTE is unable to forward the call. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'0138B506'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2040'	Inter-exchange network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2040'	Inter-exchange network
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 23

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and an invalid CPS was received by the secondary station. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'802A9670'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2036'	DCE-DSE connection
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3512' X'F059'	The connection between the calling DCE and its DSE Invalid CPS received from network
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 24

Alert Condition:

The retries have been exhausted trying to connect or reconnect a link and an unexpected CPS was received by the secondary station. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'EEEFE65A'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'F058'	Unrecognized CPS received from the network
Actions	X'3302' X'3105' X'32D0' X'82' SF X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Call Progress Signal) (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 25

Alert Condition:

This Alert indicates one of the timers, T1, T2, or T3, expired. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'92C4D92C'
Alert Type	X'01'	Permanent
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20A0' X'82' SF	No response from X.21 network (sf82) expired (Timer)
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 26

Alert Condition:

This Alert indicates timer T5 expired without receiving an expected response from the network. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'6B685EC2'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3310'	X.21 error
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20A0' X'82' SF	No response from X.21 network (sf82) expired (Timer)
Actions	X'3302' X'3105' X'32A0' X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Reporting Telephone Number)

X.21 Alert 27

Alert Condition:

This Alert indicates the calling DCE is signalling not ready. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'333F4124'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	X'0210'	Calling DCE power off
Actions	X'0200'	Check power
Install Causes	(none)	
Failure Causes	X'2050' X'3520' X'F050'	X.21 network has initiated test loop X.21 network component DCE not ready
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 28

Alert Condition:

A DCE clear indication was received by the secondary station during call establishment. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'3D061D5E'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2105'	X.21 communications/called node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3520' X'2201' X'F051'	X.21 network component Called DTE DCE not ready
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 29

Alert Condition:

A persistent DCE clear indication was received by the secondary station during call establishment. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'6332915A'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3520' X'F052'	X.21 network component Persistent DCE clear indication during call estab. (T6 exp)
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 30

Alert Condition:

A DCE controlled not ready was received by the secondary station during call establishment. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'34842630'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2105'	X.21 communications/called node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3520' X'2201' X'F053'	X.21 network component Called DTE DCE controlled not ready during call Estab.
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 31

Alert Condition:

A persistent DCE controlled not ready was received by the secondary station during call establishment. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'79B4DCE8'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3520' X'F054'	X.21 network component Persistent DCE CNR during call estab. (T6 exp.)
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 32

Alert Condition:

A DCE fault condition was received by the secondary station during call establishment. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'5E74704E'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2036'	DCE-DSE connection
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3512' X'F055'	The connection between the calling DCE and its DSE DCE fault condition during call establishment
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 33

Alert Condition:

A persistent DCE clear indication was received by the secondary station during data phase. This Alert is sent as a delayed Alert by the secondary station.

Alert ID Number		X'49DA1D89'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3311'	X.21 error — SNA secondary
Probable Causes	X'2005'	X.21 network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'3520' X'F057'	X.21 network component Persistent DCE clear indication received in data phase (T6 exp)
Actions	X'3302' X'3105' X'32C0' X'82' SF X'82' SF	If problem continues to occur then do the following Contact X.21 network information service Report the following (Calling Telephone Number) (Telephone Number Called)

X.21 Alert 34

Alert Condition:

This a delayed Alert sent by the secondary station to indicate the Cluster Controller reset.

Alert ID Number		X'C14B8A4F'
Alert Type	X'01'	Permanent (Delayed)
Alert Description	X'3313'	X.21 connection cleared
Probable Causes	X'7011'	Terminal control unit operator
User Causes	X'2310'	X.21 connection intentionally cleared by term. control Operator
Actions	X'3121'	Contact terminal control unit operator
Install Causes	(none)	
Failure Causes	(none)	

Alerts for X.25 Link Connections

This section defines the Alerts sent by X.25 nodes.

Packet Layer Control (PLC)

X.25 PLC Alert 1

Alert Condition:

A CLEAR_INDICATION packet containing a DCE-originated cause code and a diagnostic code was received by the DTE.

Alert ID Number		X'D484ED27'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006' X'2200'	Packet Layer Control X.25 Communications X.25 Network Remote Node
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20C1' X'82' SF X'82' SF X'2006' X'2200'	X.25 Communications Error -- The following indication packet was received from the network (packet type and cause code) (diagnostic code) X.25 Communications Error Remote Node
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF X'3123'	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel) Contact remote DTE's operator

Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data (Link Connect Subsystem Configuration Data) LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 2

Alert Condition:

A RESTART_INDICATION packet containing a DCE-originated cause code and a diagnostic code was received by the DTE.

Alert ID Number		X'CDA515B8'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20C1' X'82' SF X'82' SF	X.25 Communications Error -- The following indication packet was received from the network (packet type and cause code) (diagnostic code)
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 3

Alert Condition:

A RESET_REQUEST packet containing a DTE-originated cause code and a diagnostic code was sent by the DTE.

Alert ID Number		X'6A837F72'
Alert Type	X'02'	Temporary
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2200' X'2006'	Packet Layer Control X.25 Communications Remote Node X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20C2' X'82' SF X'82' SF X'2200' X'2006'	X.25 Communications Error -- The DTE sent the following request packet to the network (packet type and cause code) (diagnostic code) Remote Node X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF X'3123'	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel) Contact remote DTE's operator
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 4

Alert Condition:

A CLEAR_REQUEST packet containing a DTE-originated cause code and a diagnostic code was sent by the DTE.

Alert ID Number		X'056A9521'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008'	Packet Layer Control X.25 Communications
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20C2' X'82' SF X'82' SF	X.25 Communications Error -- The DTE sent the following request packet to the network (packet type and cause code) (diagnostic code)
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 5

Alert Condition:

A RESTART_REQUEST packet containing a DTE-originated cause code and a diagnostic code was sent by the DTE.

Alert ID Number		X'056A9521'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008'	Packet Layer Control X.25 Communications
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20C2' X'82' SF X'82' SF	X.25 Communications Error -- The DTE sent the following request packet to the network (packet type and cause code) (diagnostic code)
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 6

Alert Condition:

Time-limit T20 expired at the DTE prior to receipt of a RESTART_CONFIRMATION packet following transfer of a RESTART_REQUEST packet.

Alert ID Number		X'F50A02F0'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20D1' X'82' SF X'82' SF X'82' SF X'2006'	No response from the X.25 network -- (timer) expired (retry count) (timer setting) X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 7

Alert Condition:

Time-limit T22 expired at the DTE prior to receipt of a RESET_CONFIRMATION packet following transfer of a RESET_REQUEST packet.

Alert ID Number		X'F50A02F0'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20D1' X'82' SF X'82' SF X'82' SF X'2006'	No response from the X.25 network -- (timer) expired (retry count) (timer setting) X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional svs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 8

Alert Condition:

Time-limit T21 expired at the DTE prior to receipt of a CALL_CONNECTED or CLEAR_INDICATION packet following transfer of a CALL_REQUEST packet.

Alert ID Number		X'F50A02F0'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20D1' X'82' SF X'82' SF X'82' SF X'2006'	No response from the X.25 network -- (timer) expired (retry count) (timer setting) X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 9

Alert Condition:

Time-limit T23 expired at the DTE prior to receipt of a CLEAR_CONFIRMATION packet following transfer of a CLEAR_REQUEST packet.

Alert ID Number		X'F50A02F0'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20D1' X'82' SF X'82' SF X'82' SF X'2006'	No response from the X.25 network -- (timer) expired (retry count) (timer setting) X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

or Name=(Service Point name)
Type=X'81' (Service Point)

or Name=(Network ID)
Type=X'F6'(Network ID).

X.25 PLC Alert 10

Alert Condition:

A packet level protocol violation, on the part of the PSDN access node, was detected by the DTE. A specific diagnostic code, indicating the source and the reason for the exception, has been reported to a higher layer function.

Alert ID Number		X'BA5D4659'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'20B2' X'82' SF X'2006'	X.25 Protocol Violation Detected (diagnostic code) X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 PLC Alert 11

Alert Condition:

A DIAGNOSTIC packet, indicating a protocol violation on the part of the DTE, was sent by the PSDN or received by the DTE, or both.

Alert ID Number		X'4C323FE5'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2050' X'2008' X'2006'	Packet Layer Control X.25 Communications X.25 Network
User Causes	(none)	
Install Causes	X'8000'	Configuration Error
Actions	X'1503'	Correct Configuration
Failure Causes	X'20C3' X'82' SF X'82' SF	X.25 Communications Error -- The following diagnostic packet was received from the network (diagnostic code) (diagnostic explanation)
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

Link Access Protocol Balanced (LAPB)

X.25 LAPB Alert 1

Alert Condition:

The local station received a frame with an I-field which was too long and sent an FRMR.

Alert ID Number		X'07B1E788'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051' X'8003' X'3500'	Link Access Protocol Balanced Communication Configuration Communication Equipment
User Causes	(none)	
Install Causes	X'80C4' X'82' SF X'82' SF	Communication Configuration Error (configuration object/record) (configuration parameter)
Actions	X'1503'	Correct Configuration
Failure Causes	X'F023'	Received I-field exceeded maximum length
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 2

Alert Condition:

The local station received a frame with an invalid N(R) value and sent an FRMR.

Alert ID Number		X'C0E4E919'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051' X'3500'	Link Access Protocol Balanced Communication Equipment
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'F022'	Invalid N(R) received
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 3

Alert Condition:

The local station has retried a command frame the maximum number of times without receiving the appropiate response. The local station enters disconnected state immediately.

Alert ID Number		X'CEA222A9'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051' X'8003' X'3401' X'3541' X'3500'	Link Access Protocol Balanced Communication Configuration Local DCE Interface Cable Local DCE Communication Equipment
User Causes	(none)	
Install Causes	X'80C4' X'82' SF X'82' SF	Communication Configuration Error (configuration object/record) (configuration parameter)
Actions	X'1503'	Correct Configuration
Failure Causes	X'2006'	X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 4

Alert Condition:

An unexpected DISC command frame was received during information transfer.

Alert ID Number		X'985806E2'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051' X'3500'	Link Access Protocol Balanced Communication Equipment
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006'	X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 5

Alert Condition:

The local station sent SABM to initialize the link and received a DM frame.

Alert ID Number		X'E13004C9'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051' X'230B' X'3500'	Link Access Protocol Balanced Link Set Up Failure Communication Equipment
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006'	X.25 Communications Error
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 6

Alert Condition:

A FRMR frame was received during information transfer state. The w-bit was on in the FRMR indicating receipt of a frame from the local station which had an invalid or unknown control field.

Alert ID Number		X'00891F75'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051' X'3500'	Link Access Protocol Balanced Communication Equipment
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F010'	X.25 Communications Error Frame Reject Received: Invalid/ unsupported command or response sent
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 7

Alert Condition:

A FRMR frame was received during information transfer state. The X-bit was on in the FRMR indicating receipt of a frame from the local station which had an I-field which was not permitted.

Alert ID Number		X'CF6F806D'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051'	Link Access Protocol Balanced
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F011'	X.25 Communications Error Frame Reject Received: I-field sent when not permitted
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 8

Alert Condition:

A FRMR frame was received during information transfer state. The Y-bit was on in the FRMR indicating receipt of a frame which had an oversized I-field.

Alert ID Number		X'F5E40347'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051'	Link Access Protocol Balanced
User Causes	(none)	
Install Causes	X'80C4' X'82' SF X'82' SF	Communication Configuration Error (configuration object/record) (configuration parameter)
Actions	X'1503'	Correct Configuration
Failure Causes	X'2006' X'F013'	X.25 Communications Error Frame Reject Received: Maximum I-field length exceeded
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 9

Alert Condition:

A FRMR frame was received during information transfer state. The Z-bit was on in the FRMR indicating receipt of a frame which had an invalid N(R) specified.

Alert ID Number		X'C22CA6B4'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051'	Link Access Protocol Balanced
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F012'	X.25 Communications Error Frame Reject Received: Invalid N(R) sent
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 10

Alert Condition:

The local station received a frame with an invalid or unknown control field and sent an FRMR.

Alert ID Number		X'1F9CF04A'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051'	Link Access Protocol Balanced
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F020'	X.25 Communications Error Invalid/unsupported command or response received
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

X.25 LAPB Alert 11

Alert Condition:

The local station received a frame which had an I-field which was not permitted and sent an FRMR.

Alert ID Number		X'3FAE0180'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2051'	Link Access Protocol Balanced
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F021' X'1021'	X.25 Communications Error I-field received when not permitted Communication Controller Control Program
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address)
Additional SVs	X'52' SV X'07' SF X'05' SV X'10' SF	LCS Configuration Data LCS Link Attributes Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(adapter name) Type=X'21' (adapter) Second resource below sender: Name=(port name) Type=X'3F' (port)

Logical Link Control (LLC)

Note: For X.25 LLC protocol, protocol data units (PDUs) are the basic elements of information that are exchanged between logical link stations. For QLLC, a PDU = LLU (logical link unit), while for ELLC, a PDU = LPDU (LLC protocol data unit). When a FRMR response is mentioned in the following Alerts, it refers to an error condition of a PDU that is not recoverable, at the X.25 LLC layer, by retransmission of the identical PDU.

X.25 LLC Alert 1

Alert Condition:

The local station received a Protocol Data Unit (PDU) with an I-field which was too long and sent an FRMR.

Alert ID Number		X'6460D9A9'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052' X'8003' X'3500'	Logical Link Control Communication Configuration Communication Equipment
User Causes	(none)	
Install Causes	X'80C4' X'82' SF X'82' SF	Communication Configuration Error (configuration object/record) (configuration parameter)
Actions	X'1503'	Correct Configuration
Failure Causes	X'F023'	Received I-field exceeded maximum length
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)

Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

X.25 LLC Alert 2

Alert Condition:

The local station received a PDU with an invalid N(R) value and sent an FRMR.

Alert ID Number		X'CED07C9C'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052' X'3500'	Logical Link Control Communication Equipment
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'F022'	Invalid N(R) received
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

Note: This Alert applies only to ELLC.

X.25 LLC Alert 3

Alert Condition:

The local station has retried a command PDU the maximum number of times without receiving the appropiate response. The local station enters disconnected state immediately.

Alert ID Number		X'EAC7612A'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052' X'8003' X'2200'	Logical Link Control Communication Configuration Remote Node
User Causes	(none)	
Install Causes	X'80C4' X'82' SF X'82' SF	Communication Configuration Error (configuration object/record) (configuration parameter)
Actions	X'1503'	Correct Configuration
Failure Causes	X'2200'	Remote Node
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

X.25 LLC Alert 4

Alert Condition:

A FRMR PDU was received during information transfer state. The w-bit was on in the FRMR indicating receipt of a PDU from the local station which had an invalid or unknown control field.

Alert ID Number		X'3DA4F8CD'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052' X'3500'	Logical Link Control Communication Equipment
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F010'	X.25 Communications Error Frame Reject Received: Invalid/ unsupported command or response sent
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

X.25 LLC Alert 5

Alert Condition:

A FRMR PDU was received during information transfer state. The X-bit was on in the FRMR indicating receipt of a PDU from the local station which had an invalid I-field.

Alert ID Number		X'C15B15E8'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052'	Logical Link Control
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F011'	X.25 Communications Error Frame Reject Received: I-field sent when not permitted.
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

X.25 LLC Alert 6

Alert Condition:

A FRMR PDU was received during information transfer state. The Y-bit was on in the FRMR indicating receipt of a PDU which had an oversized I-field.

Alert ID Number		X'C8C9E4FF'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052'	Logical Link Control
User Causes	(none)	
Install Causes	X'80C4' X'82' SF X'82' SF	Communication Configuration Error (configuration object/record) (configuration parameter)
Actions	X'1503'	Correct Configuration
Failure Causes	X'2006' X'F013'	X.25 Communications Error Frame Reject Received: Maximum I-field length exceeded
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

X.25 LLC Alert 7

Alert Condition:

A FRMR PDU was received during information transfer state. The z-bit was on in the FRMR indicating receipt of a PDU which had an invalid N(R) specified.

Alert ID Number		X'CC183331'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052'	Logical Link Control
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F012'	X.25 Communications Error Frame Reject Received: Invalid N(R) sent
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

Note: This Alert applies only to ELLC.

X.25 LLC Alert 8

Alert Condition:

The local station received a PDU with an invalid or unknown control field and sent an FRMR.

Alert ID Number		X'11A865CF'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052'	Logical Link Control
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F020'	X.25 Communications Error Invalid/unsupported command or response received
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

X.25 LLC Alert 9

Alert Condition:

The local station received a PDU with an invalid I-field and sent an FRMR.

Alert ID Number		X'0283E638'
Alert Type	X'01'	Permanent
Alert Description	X'3320'	X.25 Error
Probable Causes	X'2052'	Logical Link Control
User Causes	(none)	
Install Causes	(none)	
Failure Causes	X'2006' X'F021' X'1021'	X.25 Communications Error I-field received when not permitted Communication Controller Control Program
Actions	X'3302' X'3107' X'32D0' X'82' SF X'82' SF X'82' SF X'F0A0' X'82' SF or X'82' SF	If the problem continues to occur repeatedly then do the following Contact X.25 Network Information Service Report the following (DTE address called) (DTE address calling) (local DTE address) For (locally-initiated logical channel) (remotely-initiated logical channel)
Additional SVs	X'05' SV X'10' SF	Hierarchy/Resource List Hierarchy Name List First resource below sender: Name=(port name) Type=X'3F' (port) Second resource below sender: Name=(SNA control point name) Type=X'F4' (control point)

Appendix B. Management Services Protocol Boundary Verbs

Introduction

The term *protocol boundary* is used generally to refer to the semantic definition of the data and control exchanges between two components in an SNA node. This appendix focuses on the protocol boundaries between MS and the network planners that interact with SNA/MS to request services.

These protocol boundaries are described generically here in the form of precisely defined verbs.

The MS Protocol Boundary Verbs are formally described in verb description tables and parameter descriptions. A Verb Description Table contains the primary syntax for each parameter associated with a particular verb. A Parameter Description contains a prose description of the parameter, enumeration values, and any conditions of presence associated with a particular parameter.

Verb Description Table

Column Descriptions

Supplied Parameter Name

The first column of the verb description table identifies the parameters supplied on the invocation of a particular verb, and illustrates their hierarchical relationship by indentation of the column entries.

Returned Parameter Name

The Returned Parameter Name identifies the parameters returned as a result of the invocation of a particular verb, and illustrates their hierarchical relationship by indentation of the column entries.

Parameter Reference Page (Parm Ref Page)

As primary syntax is described in a particular verb desciption table, the semantics, enumeration values, and other characteristics are described formally in the parameter description. This column contains a reference page number to where this parameter information is found.

Length

The range of length values specifies the minimum and maximum lengths of parameters which an implementation is required to accept across the protocol boundary. Sometimes the length is described as an enumeration, which may be implemented as an integer, character string, pointer, or any implementation choice.

Occurrences

Multiple occurrences of parameters may or may not be permitted. A value of "1 - <some number>" in this column indicates the allowed range of occurrences of the corresponding parameter. A value of "≥1" indicates that there is no architecturally-defined maximum. A value of "1" in this column indicates that only a single instance of the corresponding parameter is appropriate. A value of "0 - 1" indicates that an instance of the corresponding parameter is optional.

Note: An asterisk denotes special presence rules for a particular parameter. These presence rules are detailed in the corresponding parameter description.

Children

Number (Num): Each parent parameter contains a certain number of different children. This column specifies the minimum and maximum number of different children for a particular parent parameter. This column also specifies mutual exclusion among a set of optional children. If all children are optional ("0-1") and the parent parameter contains "1" for children number, one of the set of children occurs within that particular parent. This column does not account for multiple occurrences of a particular child within the parent parameter. Multiple occurrences of a particular parameter are indicated in the "Occurrences" column.

Subtable (Subtab): Sometimes the need to divide large tables into subtables becomes apparent, particularly when common parameters appear frequently within different parameter description tables. This column contains a reference page number to the page on which these common parameters are described.

Parameter Description

This description is referenced by a page number appearing in the "Parameter Reference Page" column corresponding to each parameter in the verb description table. The parameter description contains information pertaining to a particular parameter. Prose descriptions, presence rules, enumeration values and semantics associated with the corresponding entry in the verb description table may appear in the parameter description.

Protocol Boundary Verbs for File Services

VERB: Retrieve

Retrieve is issued by the network planner to request the focal point to, in turn, request another focal point or an entry point to transfer a file to the requester.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
To_Be_Fetched_Name	B-17	1-63	1	—	—
Fetching_Match_Flags	B-18	ENUM	0-1	—	—
Source_Location	—	—	1	—	B-20
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Send

Send is issued by the network planner to request the focal point to send a file to one or more focal points or entry points.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
To_Be_Fetched_Name	B-17	1-63	1	—	—
Fetching_Match_Flags	B-18	ENUM	0-1	—	—
Destruction	B-18	ENUM	1	—	—
Deleting_Match_Flags	B-18	ENUM	0-1	—	—
To_Be_Deleted_Name	B-19	1-63	0-1	—	—
Target_List	—	—	1	—	B-15
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Delete

Delete is issued by the network planner to request the focal point to delete a file at one or more focal points or entry points.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
To_Be_Deleted_Name	B-19	1-63	1	—	—
Deleting_Match_Flags	B-18	ENUM	0-1	—	—
Target_List	—	—	1	—	B-15
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Reply_To_Retrieve

Reply_To_Retrieve is issued by the focal point to report to the network planner the results of a Retrieve request.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Fetched_Name	B-19	1-63	1	—	—
Source_Location	—	—	1	—	B-20
Time_Stamp	—	—	1	—	B-16
FS_Action_Summary	B-40	—	1	—	—
SNA_Condition_Report	B-40	—	≥0	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Reply_To_Send

Reply_To_Send is issued by the focal point to report to the network planner the results of a Send request.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Stored_Name	B-17	1-63	1	—	—
Deleted_Name	B-19	1-63	0-1	—	—
Target_Location	—	—	1	—	B-21
Time_Stamp	—	—	1	—	B-16
FS_Action_Summary	B-40	—	1	—	—
SNA_Condition_Report	B-40	—	≥0	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Reply_To_Delete

Reply_To_Delete is issued by the focal point to report to the network planner the results of a Delete request.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Deleted_Name	B-19	1-63	1	—	—
Target_Location	—	—	1	—	B-21
Time_Stamp	—	—	1	—	B-16
FS_Action_Summary	B-40	—	1	—	—
SNA_Condition_Report	B-40	—	≥0	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Notification_Of_Arrival

Notification_Of_Arrival is issued by the focal point to report to the network planner that a file has arrived in an unsolicited manner from another entry point or focal point.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Stored_Name	B-17	1-63	1	—	—
Source_Location	—	—	1	—	B-20
Time_Stamp	—	—	1	—	B-16
FS_Action_Summary	B-40	—	1	—	—
SNA_Condition_Report	B-40	—	≥0	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

Protocol Boundary Verbs for Change Management

VERB: Send_and_Install

Send_and_Install is issued by the network planner to request the focal point to send a change file to one or more entry points and to install it there.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
To_Be_Fetched_Name	B-17	1-63	1	—	—
Fetching_Match_Flags	B-18	ENUM	0-1	—	—
Destruction	B-18	ENUM	1	—	—
Deleting_Match_Flags	B-18	ENUM	0-1	—	—
To_Be_Deleted_Name	B-19	1-63	0-1	—	—
Target_List	—	—	1	—	B-15
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Corequisite_Change_Name_List	—	—	0-1	—	B-14
Removability	B-25	ENUM	1	—	—
Automatic_Removal	B-25	ENUM	0-1	—	—
Pre-Test	B-26	ENUM	1	—	—
Post-Test	B-26	ENUM	1	—	—
Automatic_Acceptance	—	—	0-1	—	B-14
Activation_Use	B-28	ENUM	1	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Install

Install is issued by the network planner to request the focal point to install a change file at one or more entry points.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
Stored_Name	B-17	1-63	1	—	—
Target_List	—	—	1	—	B-15
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Corequisite_Change_Name_List	—	—	0-1	—	B-14
Removability	B-25	ENUM	1	—	—
Automatic_Removal	B-25	ENUM	0-1	—	—
Pre-Test	B-26	ENUM	1	—	—
Post-Test	B-26	ENUM	1	—	—
Automatic_Acceptance	—	—	0-1	—	B-14
Activation_Use	B-28	ENUM	1	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Remove

Remove is issued by the network planner to request the focal point to remove a change described in a change file from one or more entry points, and if successful, to delete the change file.

Removing a change means returning all components previously altered in connection with a change to their condition prior to the installation of the change.

Since the entry point remembers the list of corequisite change file names (as part of maintaining removability) the network planner need specify only one of the names in the Stored_Name parameter.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
Stored_Name	B-17	1-63	1	—	—
Target_List	—	—	1	—	B-15
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Post-Test	B-26	ENUM	1	—	—

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Accept

Accept is issued by the network planner to request the focal point to cause one or more entry points to release resources required to maintain the removability of a change described in a change file, and if successful, to delete the change file there. The resources released are, typically, unaltered versions of components affected by the change.

Since the entry point remembers the list of corequisite change file names (as part of maintaining removability) the network planner need specify only one of the names in the Stored_Name parameter.

The changes must be installed in production removably; a request to accept changes installed on trial will be rejected.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
Stored_Name	B-17	1-63	1	—	—
Target_List	—	—	1	—	B-15
DS_Priority	B-21	ENUM	0-1	—	—
DS_Security	—	—	0-1	—	B-14
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Activate

Activate is issued by the network planner to request the focal point to send a command to one or more entry points to cause reactivation of the entire entry point.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
Target_List	—	—	1	—	B-15
DS_Security	—	—	0-1	—	B-14
Force	B-28	ENUM	1	—	—
Change_Management_Activation_use	B-29	ENUM	1	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Reporting_Installation

Reporting_Installation is issued by the focal point to report to the network planner that one or more change files were (or were not) successfully installed at an entry point.

The storing of the change file may also be reported.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Stored_Name	B-17	1-63	0-1	—	—
Deleted_Name	B-19	1-63	0-1	—	—
Target_Location	—	—	1	—	B-21
Time_Stamp	—	—	1	—	B-16
FS_Action_Summary	B-40	—	1	—	—
SNA_Condition_Report	B-40	—	≥0	—	—
Detailed_Data	B-41	—	≥0	—	—
Installation_Results	B-32	—	1	2	—
Installation_Status	B-32	ENUM	1	2	—
When_Effective	B-33	ENUM	1	2	—
Reported_Change_Name_List	—	—	1	—	B-14
Pre-Test_Status	B-34	ENUM	1	—	—
Post-Test_Status	B-34	ENUM	1	—	—
Automatic_Removal_Results	B-35	—	1	2	—
Automatic_Removal_Status	B-35	ENUM	1	—	—
When_Effective	B-33	ENUM	1	2	—
Automatic_Acceptance_Status	B-35	ENUM	1	—	—
Removability_Status	B-36	ENUM	1	—	—
Activation_Use_Status	B-36	ENUM	1	—	—
Back-Level_Change_Name_List	B-24	—	0-1	2	—
Change_File_Name	B-24	1-63	1-7	—	—
Deleted_Change_Name_List	—	—	0-1	—	B-15
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Reporting_Removal

Reporting_Removal is issued by the focal point to report to the network planner that one or more change files were (or were not) successfully removed at an entry point.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Target_Location	—	—	1	—	B-21
Time_Stamp	—	—	1	—	B-16
SNA_Condition_Report	B-40	—	≥0	—	—
Detailed_Data	B-41	—	≥0	—	—
Removal_Results	B-37	—	1	2	—
Removal_Status	B-37	ENUM	1	2	—
When_Effective	B-33	ENUM	1	2	—
Reported_Change_Name_List	—	—	1	—	B-14
Post-Test_Status	B-34	ENUM	1	—	—
Secondary_Installation_Results	B-39	—	0-1	3	—
Secondary_Installation_Status	B-39	ENUM	1	—	—
When_Secondary_Effective	B-33	ENUM	1	—	—
Secondary_Activation_Use_Status	B-40	ENUM	1	—	—
Secondary_Installation_Change_Name_List	B-24	—	1	1	—
Change_File_Name	B-24	1-63	1-7	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Reporting_Acceptance

Reporting_Acceptance is issued by the focal point to report to the network planner that one or more change files were (or were not) successfully accepted at an entry point.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Target_Location	—	—	1	—	B-21
Time_Stamp	—	—	1	—	B-16
SNA_Condition_Report	B-40	—	≥0	—	—
Detailed_Data	B-41	—	≥0	—	—
Acceptance_Results	B-37	—	1	2	—
Acceptance_Status	B-38	ENUM	1	2	—
Reported_Change_Name_List	—	—	1	—	B-14
Deleted_Change_Name_List	B-24	—	0-1	—	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

VERB: Activation_Acceptance

Activation_Acceptance is issued by the focal point tc report to the network planner that an Activate request will be attempted by the entry point.

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Supplied Parameter Name					
Correlator	B-17	4	1	—	—
DS_Security	—	—	0-1	—	B-14
Target_Location	—	—	1	—	B-21
Time_Stamp	—	—	1	—	B-16
SNA_Condition_Report	B-40	—	≥0	—	—
Detailed_Data	B-41	—	≥0	—	—
Activation_Acceptance_Status	B-38	ENUM	1	2	—
Returned Parameter Name					
Return_Code	B-41	ENUM	1	—	—

Subtables

SUBTABLE: Automatic_Acceptance

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Automatic_Acceptance	B-27	—	—	1-2	—
Automatic_Acceptance_Request	B-27	ENUM	1	—	—

SUBTABLE: Corequisite_Change_Name_List

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Corequisite_Change_Name_List	B-23	—	—	1-6	—
Change_File_Name	B-24	1-63	1-6	—	—

SUBTABLE: DS_Security

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
DS_Security	B-22	—	—	2	—
DS_Security_Request	B-22	ENUM	1	—	—
DS_Security_Comp_Op	B-23	ENUM	1	—	—

SUBTABLE: Reported_Change_Name_List

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Reported_Change_Name_List	B-23	—	—	1-6	—
Change_File_Name	B-24	1-63	1-7	—	—

SUBTABLE: Deleted_Change_Name_List

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Deleted_Change_Name_List	B-24	—	—	1-6	—
Change_File_Name	B-24	1-63	1-7	—	—

SUBTABLE: Source_Location

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Source_Location	B-20	—	—	2	—
NETID	B-20	1-8	1	—	—
NAU_Name	B-20	1-8	1	—	—

SUBTABLE: Target_List

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Target_List	B-21	—	—	≥1	—
Target_Location	B-21	—	≥1	—	—

SUBTABLE: Target_Location

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Target_Location	B-21	—	—	2	—
NETID	B-20	1-8	1	—	—
NAU_Name	B-20	1-8	1	—	—

SUBTABLE: Time_Stamp

Parameter Name	Parm Ref Page	Length	Occurrences	Children	
				Num	Subtab
Time_Stamp	B-29	—	—	6-7	—
Time_Stamp_Year	B-30	1	1	—	—
Time_Stamp_Month	B-30	1	1	—	—
Time_Stamp_Day	B-30	1	1	—	—
Time_Stamp_Hour	B-31	1	1	—	—
Time_Stamp_Minute	B-31	1	1	—	—
Time_Stamp_Second	B-31	1	1	—	—
Time_Stamp_Second_Hundredths	B-32	1	0-1	—	—

Parameter descriptions

Correlator

Description:	A 4-byte binary sequence number assigned by the originator of the unit of work. A receiver of a request echoes it in the reply.
Verbs Supplied on:	Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate, Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Undefined byte string

To_Be_Fetched_Name

Description:	The FS file name used by the FS server to identify the file to be fetched.
Verbs Supplied on:	Retrieve, Send, Send_and_Install
Verbs Returned on:	None
Subtables Found in:	None
Format:	Character string

CGCSGID:	01134-00500
String Conventions:	Leading, imbedded, or trailing space characters (X'40') are not allowed.

Stored_Name

Description:	In the case of a request, the FS file name used by the entry point to identify the file that pertains to the command in the DS agent object. In the case of a report, the file name that was stored by the entry point's FS server. This parameter is present on Reporting_Installation if the FS action to store the change file is also being reported.
Verbs Supplied on:	Install, Remove, Accept, Reply_To_Send, Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, or trailing space characters (X'40') are not allowed.

Fetching_Match_Flags

Description: Matching flags to be provided the FS server on the DS protocol boundary. See the FS parameter of the same name for specification details.

Verbs Supplied on: Retrieve, Send, Send_and_Install

Verbs Returned on: None

Subtables Found in: None

Format: Enumeration

Destruction

Description: Specifies whether or not one or more files can be destroyed (overwritten) as part of the FS action requested.

Verbs Supplied on: Send, Send_and_Install

Verbs Returned on: None

Subtables Found in: None

Format: Enumeration

Possible Values

ALLOWED
NO

Deleting_Match_Flags

Description: Deleting flags to be provided to the FS server. See the FS parameter of the same name for specification details.

On Send and Send_and_Install, this parameter cannot be specified if Destruction (NO) is specified. If Destruction (ALLOWED) is specified and this parameter is omitted, then complete matching is implied.

Verbs Supplied on: Send, Send_and_Install

Verbs Returned on: None

Subtables Found in: None

Format: Enumeration

To_Be_Deleted_Name

Description:	The FS file name to be used by the FS server to identify the file to be deleted. On Send and Send_and_Install, this parameter cannot be specified if Destruction (NO) is specified. If Destruction (ALLOWED) is specified and this parameter is omitted, then the To_Be-Fetched_Name is used for matching.
Verbs Supplied on:	Send, Send_and_Install
Verbs Returned on:	None
Subtables Found in:	None
Format:	Character string

CGCSGID:	01134-00500
String Conventions:	Leading, imbedded, or trailing space characters (X'40') are not allowed.

Fetched_Name

Description:	The FS file name of the file that was fetched.
Verbs Supplied on:	Reply_To_Retrieve
Verbs Returned on:	None
Subtables Found in:	None
Format:	Character string

CGCSGID:	01134-00500
String Conventions:	Leading, imbedded, or trailing space characters (X'40') are not allowed.

Deleted_Name

Description:	The FS file name of the file that was deleted. This parameter is present on Reply_To_Send and Reporting_Installation if a file was deleted as part of the FS action.
Verbs Supplied on:	Reply_To_Send, Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, or trailing space characters (X'40') are not allowed.

Source_Location

Description: The source location of a received file, consisting of the (NETID, NAU_Name) pair.

Verbs Supplied on: Retrieve, Notification_Of_Arrival

Verbs Returned on: None

Subtables Found in: None

NETID

Description: The network ID portion of the location name.

Verbs Supplied on: Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate, Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation

Verbs Returned on: None

Subtables Found in: Source_Location, Target_Location

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, or trailing space characters (X'40') are not allowed.

NAU_Name

Description: The network-addressable unit name (e.g. logical unit name) portion of the location name.

Verbs Supplied on: Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate, Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation

Verbs Returned on: None

Subtables Found in: Source_Location, Target_Location

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, or trailing space characters (X'40') are not allowed.

Target_List

Description: The list of target locations of the requested file, consisting of (NETID, NAU_Name) pairs.

Verbs Supplied on: Send, Send_and_Install, Install, Remove, Accept, Activate

Verbs Returned on: None

Subtables Found in: None

Target_Location

Description: One of the target locations of a file, consisting of a (NETID, NAU_NAME) pair.

Verbs Supplied on: Send, Reply_To_Send, Send_and_Install, Install, Remove, Accept, Activate, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation

Verbs Returned on: None

Subtables Found in: Target_List

DS_Priority

Description: The requested DS priority.

If this parameter is not specified, the default value of DATA8 is implied.

Verbs Supplied on: Retrieve, Send, Send_and_Install, Install, Remove, Accept

Verbs Returned on: None

Subtables Found in: None

Format: Enumeration

Possible Values

DATA_12
DATA_11
DATA_10
DATA_9
DATA_8
DATA_7
DATA_6
DATA_5
DATA_4
DATA_3
DATA_2
DATA_1

DS_Security

Description:	The DS *security* service parameter indicates the security requirements for the distribution. The combination of this parameter and the DS *security_comparison_operator* yields the permitted levels of security for the distribution.
Verbs Supplied on:	Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate, Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

DS_Security_Request

Description:	Specifies the level of DS security requested.
Verbs Supplied on:	Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate, Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	DS_Security
Format:	Enumeration

Possible Values	Meaning or Usage
LEVEL1	Security is not required.
LEVEL2	Security is required.

DS_Security_Comp_Op

Description:	The DS *security_comparison_operator* parameter is used to allow a range of security service levels for a distribution. For the values it can take, refer to the DS documentation.
Verbs Supplied on:	Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate, Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	DS_Security
Format:	Enumeration

Corequisite_Change_Name_List

Description:	A list of names of change files that are to be installed by the entry point as part of the installation of the change file named in the parameter To_Be_Fetched_Name (in the case of the Send_and_Install verb) or Stored_Name (in the case of the Install verb).
Verbs Supplied on:	Send_and_Install, Install
Verbs Returned on:	None
Subtables Found in:	None

Reported_Change_Name_List

Description:	A list of names of change files that were installed, removed, or accepted by the entry point as requested.
Verbs Supplied on:	Reporting_Installation, Reporting_Removal, Reporting_Acceptance
Verbs Returned on:	None
Subtables Found in:	None

Secondary_Installation_Change_Name_List

Description:	A list of names of change files that were installed again as the result of a remove request.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None

Back-Level_Change_Name_List

Description:	A list of names of change files that were kept as back-level copies as the result of the installation, but that are now no longer installed.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None

Deleted_Change_Name_List

Description:	A list of names of change files that were deleted as the result of the installation or acceptance being reported.
Verbs Supplied on:	Reporting_Installation, Reporting_Acceptance
Verbs Returned on:	None
Subtables Found in:	None

Change_File_Name

Description:	A change file name.
Verbs Supplied on:	Send_and_Install, Install, Reporting_Installation, Reporting_Removal, Reporting_Acceptance
Verbs Returned on:	None
Subtables Found in:	Corequisite_Change_Name_List, Reported_Change_Name_List
Format:	Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, or trailing space characters (X'40') are not allowed.

Removability

Description: Specifies whether or not change files are to be installed in a removable manner (so that a subsequent Remove command can be issued against them).

If Activation_Use (TRIAL) is specified, then Removability (YES) must be specified.

Verbs Supplied on: Send_and_Install, Install

Verbs Returned on: None

Subtables Found in: None

Format: Enumeration

Possible Values

YES
DESIRED
NO

Automatic_Removal

Description: Specifies whether the entry point is to remove the change files automatically if either installation or a test fails.

Unlike a separate Remove request, the entry point does not delete the change after a successful automatic removal. This is to avoid resending changes if a simple parameter change by the network operator is sufficient to solve the problem.

This parameter is specified unless Removability (NO) is specified.

Verbs Supplied on: Send_and_Install, Install

Verbs Returned on: None

Subtables Found in: None

Possible Values	Meaning or Usage
YES	May not be specified if Removability (DESIRED) is specified.
DESIRED	
NO	

Pre-Test

Description:	Specifies whether or not the entry point is to perform a test on the change files prior to installing them.
Verbs Supplied on:	Send_and_Install, Install
Verbs Returned on:	None
Subtables Found in:	None

Possible Values
YES
DESIRED
NO

Post-Test

Description:	Specifies whether or not the entry point is to perform a test on the change files after installing or removing them.
Verbs Supplied on:	Send_and_Install, Install, Remove
Verbs Returned on:	None
Subtables Found in:	None

Possible Values

YES
DESIRED
NO

Automatic_Acceptance

Description:	Specifies whether the entry point is to accept change files automatically if installation and any tests performed are successful, in order to release resources required to maintain removability as soon as possible. Like a separate Accept request, the entry point deletes the change files after successful automatic acceptance. This parameter is specified unless Removability (NO) is specified.
Verbs Supplied on:	Send_and_Install, Install
Verbs Returned on:	None
Subtables Found in:	None

Automatic_Acceptance_Request

Description:	Specifies the type of request.
Verbs Supplied on:	Send_and_Install, Install
Verbs Returned on:	None
Subtables Found in:	Automatic_Acceptance
Format:	Enumeration

Possible Values	Meaning or Usage
YES	May not be specified if Removability (DESIRED) is specified.
DESIRED	
NO	

Activation_Use

Description:	Specifies whether the components to be altered by the installation process will be trial versions or production versions.
Verbs Supplied on:	Send_and_Install, Install
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
TRIAL	The altered components are used only for an Activate request specifying Change_Management_Activation_Use (TRIAL_AND_PRODUCTION), and supercede production components in that case.
PRODUCTION	The altered components may be used for any activation.

Force

Description:	Specifies whether or not the receiving entry point should proceed with the activation if sessions are active. (In either case, the receiver will reply with Activation Acceptance before activation is attempted.)
Verbs Supplied on:	Activate
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
NO	Reject if sessions exist to or through the control point.
YES	Activate regardless.

Change_Management_Activation_Use

Description:	Specifies which components altered by changes will be used during the activation.
Verbs Supplied on:	Activate
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
TRIAL_AND_PRODUCTION	Components altered by changes installed on trial will be used and supercede both those installed in production and also unchanged components.
PRODUCTION_ONLY	Components altered by changes installed in production will be used but not changes installed on trial.

Time_Stamp

Description:	Specifies the date and time that the request was executed by the entry point. In the case of FS reports, the date and time in the DS header (representing when Send_Distribution was issued by the entry point) is used. In the case of MS reports, the date and time of command execution is specified by the entry point in the reply CP-MSU.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	None

Time_Stamp_Year

Description:	The year.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Time_Stamp_Month

Description:	The month.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Time_Stamp_Day

Description:	The day.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Time_Stamp_Hour

Description:	The hour.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Time_Stamp_Minute

Description:	The minute.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Time_Stamp_Second

Description:	The second.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Time_Stamp_Second_Hundredths

Description:	Hundredths of a second.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	Time_Stamp
Format:	Unsigned decimal integer

Installation_Results

Description:	Specifies the status of installation and when it is effective.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None

Installation_Status

Description:	Specifies whether or not installation was successful.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

When_Effective

Description:	Specifies when the changed components will be in use.
Verbs Supplied on:	Reporting_Installation, Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
IN_USE	Changed components are now in use.
ACTIVATION_REQUIRED	Components are changed, but activation is required to put them in use.
NOT_APPLICABLE	Not applicable, because the request was not attempted.

When_Secondary_Effective

Description:	Specifies when the changed components will be in use after secondary installation.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
IN_USE	Changed components are now in use.
ACTIVATION_REQUIRED	Components are changed, but activation is required to put them in use.

Pre-Test_Status

Description:	Specifies whether or not the pre-test was successful.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

Post-Test_Status

Description:	Specifies whether or not the post-test was successful.
Verbs Supplied on:	Reporting_Installation, Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

Automatic_Removal_Results

Description:	Specifies the status of automatic removal and when it is effective.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None

Automatic_Removal_Status

Description:	Specifies whether or not automatic removal was successful.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

Automatic_Acceptance_Status

Description:	Specifies whether or not automatic acceptance was successful.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

Removability_Status

Description:	Specifies whether the change file was installed removably or not. An entry point does not make a decision of which way to install a change file independent from the focal point request. However, this status information is returned in the report.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
INSTALLED_REMOVABLY	
INSTALLED_NONREMOVABLY	
NOT_INSTALLED	The installation was not successful.

Activation_Use_Status

Description:	Specifies whether the change file was installed on trial or in production. An entry point does not make a decision of which way to install a change file independent from the focal point request. However, this status information is returned in the report.
Verbs Supplied on:	Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
TRIAL	
PRODUCTION	
NOT_INSTALLED	The installation was not successful.

Removal_Results

Description:	Specifies the status of removal and when it is effective.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None

Removal_Status

Description:	Specifies whether or not removal was successful.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

Acceptance_Results

Description:	Specifies the status of acceptance.
Verbs Supplied on:	Reporting_Acceptance
Verbs Returned on:	None
Subtables Found in:	None

Acceptance_Status

Description:	Status.
Verbs Supplied on:	Reporting_Acceptance
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values

SUCCESSFUL	
NOT_SUCCESSFUL	
NOT_ATTEMPTED	Not attempted and will not attempt.

Activation_Acceptance_Status

Description:	Specifies whether or not activation will be attempted.
Verbs Supplied on:	Activation_Acceptance
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values

WILL_ATTEMPT

WILL_NOT_ATTEMPT

Activation_Status

Description:	Specifies whether or not activation was successful.
Verbs Supplied on:	Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values

SUCCESSFUL

NOT_SUCCESSFUL

Change_Management_Activation_Use_Status

Description:	Specifies whether trial components were used, or not used, in the activation. An entry point does not make a decision of which way to activate itself, independent from the focal point request. However, this status information is returned in the report.
Verbs Supplied on:	Reporting_Activation
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
TRIAL_AND_PRODUCTION	
PRODUCTION_ONLY	
NOT_APPLICABLE	The activation was not successful.

Secondary_Installation_Results

Description:	Specifies that one or more change files were installed as the result of a removal of other change files.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None

Secondary_Installation_Status

Description:	Specifies that secondary installation was successful.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values

SUCCESSFUL

Secondary_Activation_Use_Status

Description:	Specifies whether secondary installation was on trial or in production.
Verbs Supplied on:	Reporting_Removal
Verbs Returned on:	None
Subtables Found in:	None
Format:	Enumeration

Possible Values	Meaning or Usage
TRIAL	
PRODUCTION	

FS_Action_Summary

Description:	SNA/FS action summary, as defined by SNA/FS.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation
Verbs Returned on:	None
Subtables Found in:	None

SNA_Condition_Report

Description:	SNA Condition Report, as defined by SNA/FS.
Verbs Supplied on:	Reply_to_Retrieve, Reply_to_Send, Notification_of_Arrival, Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation This parameter is present once for each condition report returned in either the agent object or server object.
Verbs Returned on:	None
Subtables Found in:	None

Detailed_Data

Description:	Reports conditions unique to entry-point implementations.
Verbs Supplied on:	Reporting_Installation, Reporting_Removal, Reporting_Acceptance, Activation_Acceptance, Reporting_Activation This parameter is present once for each such item returned in either the agent object or server object.
Verbs Returned on:	None
Subtables Found in:	None

Return_Code

Description:	The results of verb execution.
Verbs Supplied on:	None
Verbs Returned on:	Retrieve, Send, Send_and_Install, Install, Remove, Accept, Activate
Subtables Found in:	None
Format:	Enumeration

Possible Values

OK
FS_ORIGIN_EXCEPTION
DS_ORIGIN_EXCEPTION

Appendix C. SNA/FS File Names Defined by SNA/MS

The total length of all ten tokens is restricted to (64-n+1) bytes, where n = number of tokens specified. This is a practical constraint, not the natural architectural limit.

Tokens shown as product specifiable (may, but not must, be specified), using SNA/FS guidelines (for example, regarding hierarchical levels of authority).

Implementations may choose to assign more stringent matching requirements when they create files, but not less stringent ones. For example, an implementation creating a file may assign an attribute of MUST_MATCH to a token described in the architecture as having an attribute of NEED_NOT_MATCH (but not the other way around).

Table C-1. Identification tokens for microcode

Token number	Token Attributes (assigned when file is created)	Contents
1	Must match, not generable, type unspecified	C'MCODE'
2	Must match, not generable, type unspecified	Machine type (4 characters today)
3	Need not match, not generable, type unspecified	Model number (3 characters today). If none assigned, or if this change is for all model numbers, C'NA'.
4	Need not match, not generable, type unspecified	One of the following: • C'PATCH' • C'MCF' • C'SUFFIXEC' • C'MAINTEC' • C'FUNCTEC' • C'FEATURE'
5	Need not match, not generable, type unspecified	Microcode change name (8 characters today; e.g.: patch or MCF number, EC level)
6 - 10	Product specifiable	Product specifiable

Table C-2. Identification tokens for microcode customizing data

Token number	Token Attributes (assigned when file is created)	Contents
1	Must match, not generable, type unspecified	C'MCUST'
2	Must match, not generable, type unspecified	Machine type (4 characters today)
3	Need not match, not generable, type unspecified	Model number (3 characters today). If none assigned, or if this change is for all model numbers, C'NA'.
4	Need not match, not generable, type is NETID	NETID portion of location name of node for which this customizing data was created. If not applicable, C'NA'.
5	Need not match, not generable, type is LUNAME	LUNAME portion of location name of node for which this customizing data was created. If not applicable, C'NA'.
6	Need not match, not generable, type unspecified	Product defined, but must be specified to ensure uniqueness
7 - 10	Product specifiable	Product specifiable

Acronyms and Abbreviations

ACTLU	activate logical unit
ACTPU	activate physical unit
BF	boundary function
CD	change direction
CNR	carrier-to-noise ratio
CP	control point
CPMS	control point management services
CP-MSU	control point management services unit
CRC	cyclic redundancy check
CSMA/CD	carrier sense multiple access with collision detection
CTS	clear to send
DCE	data circuit-terminating equipment
DLC	data link control
DS	distribution services
DSE	data service element
DSU	distribution service unit
DTE	data terminal equipment
EB	end bracket
EC	engineering change
EP	entry point
FM	function management
FP	focal point
FRMR	frame reject
FS	file services
GDS	general data stream
GMT	Greenwich Mean Time
ID	identification
IML	initial microprogram load
IPL	initial program load
LAN	local area network
LAPB	link access procedure, balanced
LCS	link connection subsystem
LLC	logical link control
LMS	local management services
LU	logical unit
MAC	medium access control
MCF	microcode fix
MES	miscellaneous equipment specification
MS	management services
MU	message unit
MV	major vector
NAU	network addressable unit
NETID	network identification
NMVT	Network Management Vector Transport
NRM	normal response mode
NS	network services
NSD	National Service Division
OEM	original/other equipment manufacturer
PRID	procedure related identifier
PSID	product set identification
PU	physical unit
PUMS	physical unit management services
QPI	query product identification
RS	ring station
RSP	response
RTM	response-time monitor
RTS	request to send
RU	request-response unit
SAA	Systems Application Architecture
SDLC	Synchronous Data Link Control
SF	subfield
SNA	Systems Network Architecture
SNACR	SNA Condition Report
SNA/DS	SNA/Distribution Services
SNA/FS	SNA/File Services
SNA/MS	SNA/Management Services
SNRM	set normal response mode
SSCP	system services control point
SV	subvector
TP	transaction program
VAR	value added remarketer

Glossary

This glossary defines terms used in this book. It should be considered an adjunct to the glossary in *SNA Concepts and Products*, GC30-3072, and in *Dictionary of Computing*, SC20-1699.

A

accept. To cancel removability of a change (see Install). Releases resources required to maintain the removability of a change.

Alert. A record sent to a control point to communicate the existence of an Alert condition.

Alert condition. A problem or impending problem for which some or all of the process of problem determination/diagnosis/resolution is expected to require action at a control point.

B

C

change. An alteration (addition, deletion, or modification) of one or more information system components, of one of the following types: hardware (may include microcode), or software (system or application). The term *change* also refers to an SNA/File Services data object containing microcode, microcode customizing data, software, software customizing data, applications data, procedures, or documentation.

change management. The process of planning (e.g., scheduling) and controlling (e.g., distributing, installing, and tracking) changes in an SNA network.

change management plan. A plan is a sequence of change management requests to be performed by a focal point with regard to one or more entry points. It is prepared by the network planner.

configuration management. The control of information necessary to identify both physical and logical information system resources and their relationship to one another.

corequisite change file. A change file that must be installed together with another change file, as part of the same process.

control point management services (CPMS). A component of a control point, consisting of management services function sets, that provides facilities to assist in performing problem management, performance and accounting management, change management, and configuration management. Capabilities provided by the CPMS include sending requests to a physical unit management services (PUMS) to test system resources, collecting statistical information (e.g., error data, performance data) from a PUMS about the system resources, and analyzing and presenting test results and statistical information collected about the system resources.

D

default/replacement code point. A two-byte code point in which the first byte indexes text providing a high-level description of a type of condition and the second byte indexes text providing a more specific description of a particular condition of that type.

delayed Alert. An Alert reporting an Alert condition that prevented the Alert sender from sending that Alert to a control point. The fact that a delayed Alert is sent implies that the Alert condition it reports is no longer impacting availability. *See also* held Alert.

delete. In the context of SNA/MS, a protocol boundary verb to delete a change file.

E

engineering change (EC). A change to hardware or microcode to correct one or more design defects, or to modify the design. There are three types of ECs: *suffix, maintenance, and functional.*

entry point. An entry point is an SNA node that provides distributed network management support. It may be a type 2.0 or type 2.1 node. It sends SNA-formatted network management data about itself and the resources it controls to a focal point for centralized processing, and it receives and executes focal point initiated requests to manage and control its resources.

error. The smallest detectable anomaly or exception that can occur in an information system. Errors may be caused by hardware, software, microcode, media, or external causes, e.g., people or environmental abnormalities.

error condition. A state whose existence is indicated by (1) the occurrence of an error, and, possibly, (2)

additional specific events, often errors resulting from attempts to recover from the initial error. Associated with each type of error is a recovery procedure to be followed when an error of that type occurs. The results of this recovery procedure determine whether the error condition is to be classified as *recoverable* or *irrecoverable*. An error condition is classified as *irrecoverable* if the operation being attempted when the error was detected is not completed without the use of recovery techniques external both to the component detecting the error, e.g., a hardware adapter, and to its controlling component, e.g., microcode controlling the adapter. A *recoverable* error condition is one in which the operation being attempted at the time the error was detected is completed by using a recovery procedure internal either to the component detecting the error or to its controlling component.

F

feature change. A change to hardware (or microcode) to add new capabilities. Sometimes called a *sales change*.

firmware. *See* microcode.

focal point. A focal point is an entry point that provides centralized management and control for other entry points for one or more network management categories.

function set. *See* management services function set.

function set group. *See* management services function set group.

functional EC. An EC that enhances reliability/availability/serviceability (RAS) or adds new capabilities (in addition to correcting one or more design defects).

G

general function set. *See* general management services function set.

general management services function set. A management services function set that transports management services data or provides a database function for this data in a control point.

H

hardware. Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation.

held Alert. An Alert that an Alert sender was unable to send to a control point immediately. The sending of a held Alert implies nothing about whether the Alert condition it reports is still impacting availability. *See also* delayed Alert.

I

impending problem. An error condition that has the potential for resulting in a loss of availability of a system resource to an end user.

indicated resource. The SNA or non-SNA entity most closely associated with an Alert condition.

install. In the context of SNA/MS, to alter all network components necessary to effect a change. A change may be installed removably or nonremovably.

L

local management services (LMS). A component that provides for collection and exchange of management data with the physical unit management services component regarding resources associated with the node. Local management services components exist in a logical unit, path control manager, DLC manager, and physical resource manager.

M

machine level control (MLC). A complete history of engineering and feature changes to a hardware product. This term is often misused to describe the entire range of activities involved in processing a user's order for a machine.

maintenance EC. An EC correcting one or more hardware or microcode design defects. Usually corrects more defects than a suffix EC and has undergone more testing.

management services. *See* SNA/MS.

management services function set. A collection of specific services that together perform an overall management services function for a physical unit or a control point (e.g., Alert). Function sets are composed of a base and optional subsets.

management services function set group. The components or subset of components providing the function described by the corresponding function set.

microcode. Microinstructions used in a product as an alternative to hard-wired circuitry to implement functions of a processor or other system component. (*See* microinstruction.)

microcode fix (MCF). A change to correct a single microcode design defect. Higher quality than a *patch*, and intended for broad distribution. Considered functionally equivalent to a software PTF.

microinstruction. An instruction that controls data flow and sequencing in a processor at a more fundamental level than a machine instruction.

miscellaneous equipment specification (MES). A request from marketing that results in a feature change. Also, the procedure to install the feature change.

N

NETWORK MANAGEMENT VECTOR TRANSPORT (NMVT). A management services RU that flows over an active session between the physical unit management services component and the control point management services component of a control point.

network operator. A person or program responsible for controlling the operation of all or part of a network.

network management. The process of planning, organizing, and controlling a communication-oriented data processing or information system.

network planner. A person or program responsible for planning the configuration (and changes) of all or part of a network.

P

patch. A change to correct a single defect, in hardware, microcode, or software. Temporary in nature, usually applied in *corrective mode*, i.e. to solve a problem that has actually been experienced at the site installing the patch.

performance and accounting management. The process of quantifying, measuring, reporting, and controlling the responsiveness, availability, utilization, and costs of an information system.

physical unit management services (PUMS). A component of the PU, consisting of management services function sets, that provides management services for a node and its associated resources. PUMS is dependent on control point management services (CPMS) for providing some information and control (e.g., configuration knowledge of the connection between itself and its control points). Capabilities provided by the PUMS include receiving requests from a CPMS, passing these requests to the appropriate local management services (LMS) for processing, and sending data to a CPMS.

prerequisite change file. A change file that must be installed prior to the subsequent installation of another change file.

problem. An error condition resulting in a loss of availability of a system resource to an end user.

problem bypass and recovery. The process of providing partial or complete circumvention of a problem, usually prior to the final resolution of the problem.

problem determination. The process of isolating a problem to the failing hardware device, software product, microcode component, medium, or external cause in order to identify the organization responsible for problem diagnosis.

problem diagnosis. The process of determining the precise cause of a hardware, software, microcode, medium, or externally caused problem and the precise action required to resolve the problem.

problem management. The management discipline that manages a problem from its detection through it final resolution. Problem management is composed of the following:

Problem determination
Problem diagnosis
Problem bypass and recovery
Problem resolution
Problem tracking and control

problem resolution. The process of taking action to correct the error condition detected as a problem or impending problem.

problem tracking and control. The process of tracking a problem until its resolution.

production. An installation option that results in altered components being installed for production use. Such components will be activated for any activation request pertaining to them.

program. A set of actions or instructions that a machine is capable of interpreting and executing. In the context of SNA/MS, there are two types of programs: software and microcode.

R

request for price quotation (RPQ). A feature change that is created specially for a limited set of users. Popular RPQs become orderable feature changes. RPQs are assigned numbers.

remove. In the context of SNA/MS, to return all components previously altered in connection with a change to their condition prior to the installation of the change.

retrieve. In the context of SNA/MS, a protocol boundary verb to retrieve a change file.

S

send. In the context of SNA/MS, a protocol boundary verb to send a change file.

SNA/Management Services (SNA/MS). The services provided to assist in management of SNA networks.

software. Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system.

In the context of SNA/MS, a special kind of software, microcode, is defined (*see* microcode, program.)

source update. A change to programming source statements (prior to assembly).

specialized function set. *See* specialized management services function set.

specialized management services function set. A management services function set that analyzes data for a particular management services function. In a control point, it also interacts with a network operator.

suffix EC. An EC correcting one or more hardware or microcode design defects. (*See* Maintenance EC.)

system services control point (SSCP). A control point within a subarea network for managing the configuration, coordinating network operator and problem determination requests, and providing directory support and other session services for end users of the network. Multiple SSCPs, cooperating as peers with one another, can divide the network into domains of control, with each SSCP having a hierarchical control relationship to the physical units and logical units within its own domain.

system services control point domain. The system services control point and the physical units (PUs), logical units (LUs), links, link stations and all the resources that the SSCP has the ability to control by means of activation requests and deactivation requests.

T

test. In the context of SNA management services, to exercise as many altered components as feasible to observe whether a change has produced the desired result (typically, an absence of error conditions).

trial. An installation option that results in altered components superseding components installed for production use. Such components can only be activated if use of both trial and production components is specified in the activation request.

type 2.0 Node. An SNA node that attaches to a subarea boundary function and receives its management services support via an SSCP-PU session.

V

vital product data (VPD). Information imbedded in a product that is in machine-readable form and will be used to uniquely identify a product or product instance or to provide other information.

Note: The information is *vital* only in the context of how it is used (for example, to determine service entitlements, or by specific SNA management services categories like problem or change management). There may be other data describing the product kept by a configuration management focal point that is equally *vital* to some category, but is not kept by the product and so does not fall within this definition.

Index

N

P

Q

R

S

U

V

Printed in U.S.A.

Reader's Comment Form

Systems Network Architecture
Management Services
Reference

Publication No. SC30-3346-2

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: Copies of IBM Publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are: clarity, accuracy, completeness, organization, coding, retrieval, and legibility.

Comments: _____

_____

_____

_____

_____

_____

_____

What is your occupation? _____

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

SC30-3346-2

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

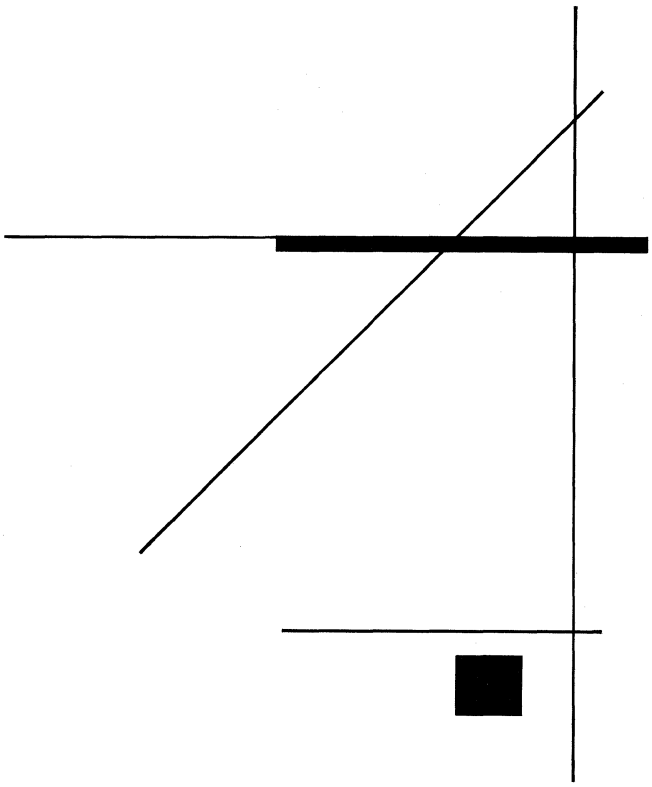
International Business Machines Corporation
Dept. E70
P.O. Box 12195
Research Triangle Park, N.C. 27709-9990

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

Publication Number
SC30-3346-2

Printed in USA



SC30-3346-2