Bull DPX/20

OSI Services
Reference Manual

AIX

ORDER REFERENCE

86 A2 05AQ 02

Bull DPX/20

OSI Services
Reference Manual

AIX

Software

June 1996

Bull Electronics Angers S.A.
CEDOC
Atelier de Reprographie
331 Avenue Patton
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 05AQ 02

Printed in France

Suggestions and criticisms concerning the form, content, and presentation of this book are invited. A form is provided at the end of this book for this purpose.

Trademarks and Acknowledgements

About This Book

The OSI Services Reference Manual supplies users with detailed information about how to install, configure and manage the OSI Stack. The following information is provided:

- Overview of the OSI Stack V2 for AIX.
- Introduction to the OSI model.
- OSI Stack characteristics (functions, addressing and telecommunications profiles).
- Guidelines for installation planning, installation, configuration and use of networks, including relays.
- Security management.
- Configuration tools (SMIT and commands).

Audience

This manual addresses an audience of:

- UNIX system administrators who have to install and configure the OSI Stack and define an OSI network.
- administrators who have to solve interoperability problems in heterogeneous environments.
- application developers who must be aware of OSI Stack features.

It is assumed that you are familiar with data communications concepts, including the Open System Interconnection (OSI) 7–layer model and OSI addressing.

Those requiring detailed information on UNIX operating systems and the 'OSI for UNIX' offer should refer to the Bibliography on page B-1, for further reading.

Operating System Level

This document is at Revision 02 level, which applies to AIX Version 4.1.

How to Use This Book

Reader Profile	Chapter
Novice in OSI	1 & 5
New to OSI Stack installation on DPX/20	7 & 8
Expert needing to fine–tune the system	4 & 11
Administrator familiar with OSI	2, 4 & 6
Administrator with system interoperability problems	2, 3, 4, Appendix A
Administrator needing configuration tools	9, 10, 11 & Appendix C
Trouble–shooter	OSI Diagnostic Interactive Toolkit (ODIT) User's Guide
Administrator familiar with OSI Stack V1	OSI Communications Porting Guide

Document Overview

This manual is structured in 11 chapters, four appendices, a glossary and a general index:

Chapter 1	**Introducing OSI Communications** Provides a description of the OSI Model, the OSI Stack and its architecture.
Chapter 2	**OSI Stack Characteristics** Provides an overview of OSI Stack functions and profiles.
Chapter 3	**Addressing** Addressing in detail.
Chapter 4	**Configuration & Limits** Tabulates configurable parameters.
Chapter 5	**Planning Your Installation** Advises how to plan your installation.
Chapter 6	**Supporting Your Network Topology** Using LAN and X.25 networks.
Chapter 7	**Installing the OSI Stack** Installing the Stack using SMIT.
Chapter 8	**Configuring, Starting and Stopping the OSI Stack.** Basic Stack operations and configuration guide–lines, with worksheets.
Chapter 9	**Configuring the OSI Stack Using SMIT.** Managing the Stack using SMIT.
Chapter 10	**Configuring the OSI Stack Using Commands.** Managing the Stack using commands.
Chapter 11	**OSI Stack Toolkit** Using management tools.
Appendix A	**OSI Telecommunications Profiles** Presents the various connection profiles available for the OSI Stack.
Appendix B	**Bibliography** Bibliography of further source information.
Appendix C	**OSI Configuration Error Codes and Messages** Lists error codes and messages returned by the configurator.
Appendix D	**OSI Applications & Communications Adapters Cross Reference** Lists the OSI Applications supported on each type of Communications Adapter.
Glossary	Alphabetical list of terms and abbreviations used in this manual.
Index	General index.

Conventions Used in This Document

The following typographic conventions are used in this document:

Bold	Bold characters are used to highlight key words, commands and subroutines.
Italic	Italic characters represent file names and user supplied variables.
`Courier`	Courier characters are used in examples and for user commands entered on the terminal keyboard and for examples.
Fixed pitch	Fixed pitch characters are used to present system display information.
Display fields	Individual fields on a display screen are outlined and presented sequentially.
*** Mandatory**	The names of fields a user must complete are presented in bold type with an asterisk (*) to the left.
#	Numeric field.
X	Hexadecimal field.
< >	Angle brackets enclose a user–supplied variable.

References to Standards

Applicable Standards are listed in the Bibliography on page B-1.

Bibliographical References

Reference documents, cited in the text, are listed in the Bibliography on page B-1.

Terminology

The term “machine” is used to indicate the proprietary hardware, in this case the DPX/20 family of single– and multi–processors.

The term “Operating System” is used to indicate the proprietary operating system software, in this case AIX.

Revision 02 Modifications

Updates include:

- Dynamic removing of NSAPs and adapters.
- NSAPs have now a type (local/group). Their maximum number is 32.
- More transport parameters can be modified.
- clnp_resolver is now called osiribd.
- osisubnet command is removed and embedded in osiadapt and osisnapshot commands.
- trmem command is removed and embedded in trmad command.

To help you access information:

- An index, listing the **management functions**, together with configurator menu FastPath and document reference pages is provided before the first chapter.
- All **examples** are indexed.

To help you prepare your OSI sites, the following **worksheets** are provided, in Chapter 8:

- Station Worksheet, on page 8-43.
- Calling Application Worksheet, on page 8-45.
- Relay Worksheet, on page 8-47.
- Called Application Worksheet, on page 8-51.

Table of Contents

Worksheets for OSI SItes

The following worksheets are provided to help you prepare your OSI sites:

Parameter Management – Quick Reference

OSI Stack management functions listed in alphabetical order, together with configurator menu FastPaths and document page references.

A

C

D

E

F

I

L

M

N

O

P

R

S

T

X

Chapter 1. Introducing OSI Communications

Introducing OSI Communications Overview

This section introduces the Distributed Computing Model (DCM), the Open Systems Interconnection (OSI) Communications Stack and describes how the Stack is implemented using the OSI Layer Model and the OSI Configurator.

You can find more information in:

- Introduction, on page 1-1.
- Layer Structured Model, on page 1-3.
- Layer Services, on page 1-4.
- Naming and Addressing, on page 1-6.
- Network Interconnections, on page 1-7.

Introduction

Bull has defined the Distributed Computing Model (DCM) to cope with complex and multiple networks including systems from different vendors. The DCM covers three major networks:

- TCP/IP, the defacto standards for UNIX communications
- OSI, the emerging international standard
- SNA, the defacto standards for IBM mainframe connectivity.

The DCM addresses three types of users: application developers, system administrators and end users.

The OSI offer for UNIX is integrated in the DCM and the OSI Stack is the basic building block of this offer. It addresses all types of users identified by the DCM:

- applications developers using programming interfaces over X.25, transport, session and ACSE/Presentation services.
- system administrators of the OSI Stack, either with specific stack tools (local administration), or through the Integrated System Management (ISM, remote administration).
- end users with mailing applications (X400), or Directory Service (X500), or File Transfer (FTAM, UFT), or Remote Terminal Facilities (TPAD/HPAD, OTM, OSI–VT), OSI–TP or Data Base Access (SQL*NET) also use the OSI Stack.

The OSI Stack coexists with the TCP/IP and SNA communication frameworks and, with the NetShare product, is able to use OSI upper layers over TCP/IP.

The OSI Stack is split into two independent parts:

- lower layers (up to the transport layer), supporting Ethernet, Token Ring, FDDI and X.25 connections.
- upper layers (session, presentation and ACSE).

The whole of the OSI Stack fully satisfies International standards as well as profiles and agreements defined by CEN/CENELEC in Europe, NIST in the United States and Government profiles (US–GOSIP, UK–GOSIP).

The links between the OSI Stack and other components of OSI for the UNIX offer is shown in Figure 1.

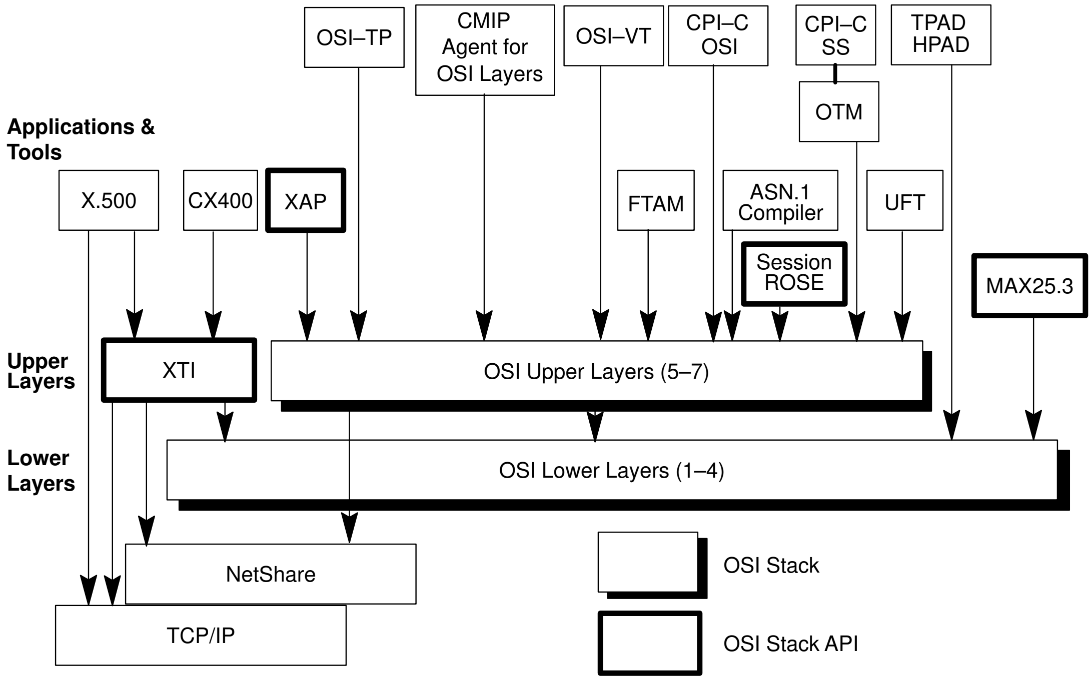


Figure 1. Showing Links between OSI Stack and other OSI Components.

About OSI Communications

This section provides an overview of the OSI communications; it is not a description of the OSI product features.

Layer Structured Model

Open Systems Interconnection is the reference model, defined in ISO 7498, for interconnecting systems from different manufacturers in which communications functions are divided into a vertical set of seven processing layers. Each layer performs a related subset of the functions required to communicate with another system, relying on the next lower layer to perform more primitive functions and in turn, providing services to the next higher layer.

In order for two (or more) systems to communicate, there must be the same sets of layered functions present in all the systems. A peer protocol enables communication to be achieved between corresponding entities in the same layer in two different systems.

The OSI Layer Structured Model in Figure 2 shows applications X and Y communicating between two systems. In order for application X to send a message to Y it must first invoke the application layer (7). This layer establishes a peer relationship with layer 7 of System B, using layer 7 protocol. This operation requires services from layer 6 and so both layers 6 communicate, down through the layers, using their own protocol. Layer 6 demands services from layer 5 and so on until each layer is communicating with its opposite number.

Finally the physical layer (1) passes the data from each layer through a transmission medium to the other system.

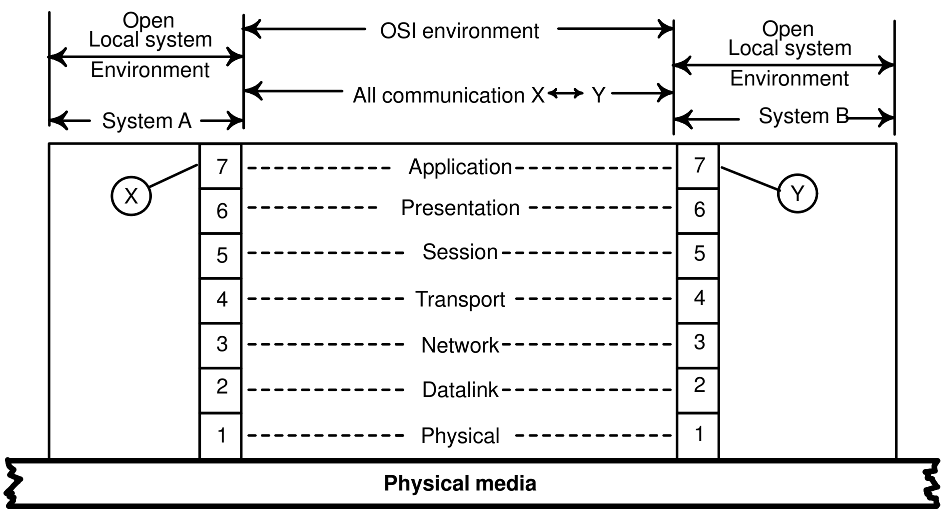


Figure 2. The OSI Layer Structured Model.

This layer structure is summarized below.

Layer Services

Each layer has a well–defined role providing well–defined services.

Layer 7 – Application Layer and Association Control Service Element (ACSE)

The application layer contains a wide variety of services (called ASE, Application Service Elements) such as Message Handling Service (MHS), File Transfer (FTAM), Virtual Terminal (VT), Transactional Processing (TP), Remote Operation (ROSE), Reliable Transfer (RTSE), Management Protocol (CMIP), ...

ACSE is a CASE (Common Application Service Element) which provides basic facilities for the control of an application association between two application entities that communicates by means of a presentation connection.

Layer 6 – Connection Oriented Presentation Service Provider

The presentation layer is concerned with the representation of information in transit between open systems. The data values are represented using an abstract syntax. To communicate successfully, two application–entities must have an agreement on the set of abstract syntaxes they intend to use. The presentation service is divided into five facilities (connection establishment, connection termination, context management, information transfer, dialog control) and some of them (information transfer, dialog control) contain services that have the same semantic that the equivalent services of the session provider.

Layer 5 – Connection Oriented Session Service Provider

The session service provides the means for organized and synchronized exchange of data between cooperating Session Service users. It establishes connections and release them in an orderly manner. It allows half–duplex and duplex data exchange. Synchronization points are used to resume the dialog from an agreed point.

Layer 4 – Transport Service Providers

The transport service provides a transparent transfer of data between two transport service users. The transport service assumes the independence with the resources of the lower communication services (same quality of service with various type of network layers), the end to end data transfer, a transparent data transfer, and should optimize the use of the communication resources to reduce the cost of the quality of service required by the transport service user. A connectionless transport service is also defined.

The transport protocol is able to use either a connectionless or connection oriented underlaying service. Several classes are defined for the connection oriented transport protocol:

- class 0 offers a minimal service. There is no connection establishment retry on error detection, and neither multiplexing nor flow control.
- class 1 provides retry for error notified by the underlaying connection oriented service provider.
- class 2 offers multiplexing and flow control mechanisms over a connection oriented underlaying service.
- class 3 offers class 1 and class 2 services.
- class 4 is an extension of class 3 which also retry for error not notified by the underlaying service but detected by the transport provider itself.

Layer 3 – Connection Oriented Network Service

The connection oriented network layer provides means to establish maintain, and terminate network connection between open systems. It hides everything about the underlaying data transmission and switching technologies used to connect the systems.

The addressing is based on NSAPs, which are defined in ISO 8348/AD2.

CONS can be offered on top of the X25.3 packet layer protocol using specific mechanisms to transmit NSAPs (as detailed in ISO 8878). X25.3 packet layer protocol is also an ISO standard connection–orientated protocol, but does not support NSAP addressing.

Layer 3 – ConnectionLess Network Service Provider (CLNP)

The ConnectionLess network service provides to its users independence of routing and relay consideration, and a uniform appearance of the services over homogenous or heterogeneous sets of interconnected subnetworks.

The addressing is based on NSAPs, which are defined in ISO 8348/AD2.

Layer 3 – Routing Framework

The routing framework is defined in ISO TR 9575, 89. Its aim is to maintain the routing information used by CLNP in order to allow NPDUs to flow between End Systems. It is based on the concept of End System that off–load the routing functions to Intermediate Systems and have fewer routing choices, and Intermediate Systems which are in charge to solve the off–loaded routing functions.

The routing procedures used are the static routing and the distributed adaptive routing. The adaptive routing is based on dynamic routing protocols like ES–IS (ISO 9542, 88) and IS–IS (ISO 10589, 92) protocols. The system configuration and the ES–IS protocol are working in conjunction with the CLNP provider to allow management of the routing information in the following way:

- system configuration allows to by–pass dynamic routing protocols, and to define off–line optimized routing paths.
- dynamic Routing Information is centralized in Intermediate Systems; the ES–IS protocol provides a mean for Intermediate System to dynamically handle relevant information to End systems located on the same subnetwork and for End System to have a knowledge of Intermediate Systems.

The Routing Information Base is the collection of information either defined by system configuration, or dynamically gathered by routing protocols (ES–IS, IS–IS, for example). It is used by the decision process to create the Forwarding Information Base: this base enables to find, from a remote NSAP, the sub–network address of the next system, and the local sub–network address to reach the next system involved in the PDU routing. This system is either an intermediate system or the destination End System itself.

Layer 2 – Data Link

The data link permits the transfer of blocks of information across the physical link, with the required synchronization, error detection and flow control.

Layer 1 – Physical

Handles the mechanical, electrical, functional and procedural characteristics to access and transmit the bit stream over the physical medium.

Naming and Addressing

A more detailed look at the layer concept and OSI terminology is given in Figure 3. Any layer is referred to as the N layer. The names of constructs in that layer also have the prefix N. An N entity implements functions of the N layer and the protocol for communicating with N entities in other systems.

Within a system there can be one or more entities in each layer. At each layer, entities can be identical or different; for example, if there are different protocols at that level.

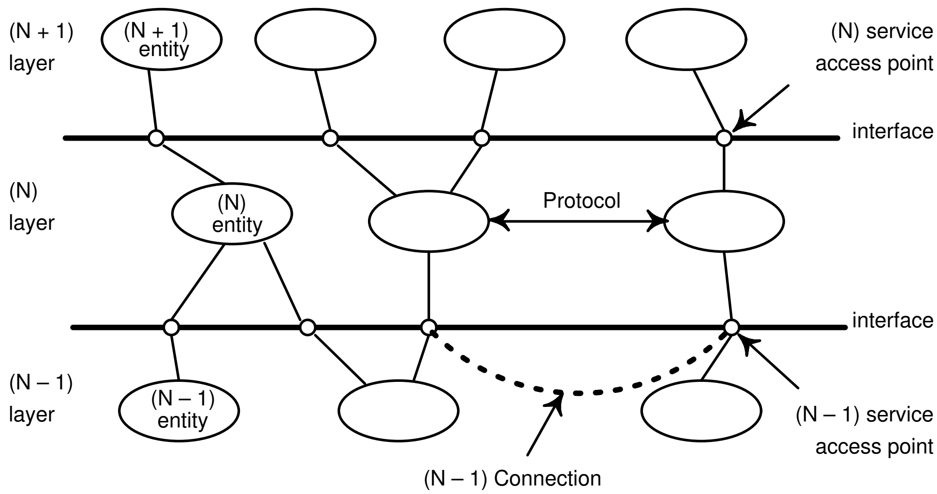


Figure 3. OSI Layer Concept.

Communication between entities in adjacent layers is across an interface, made up of one or more Service Access Points (SAPs).

Services demanded by one layer (N) to the layer below it (N–1) are achieved using primitives, specifying the function to be performed and containing control information and data. Data passes between entities in the form of Data Units.

A Protocol Data Unit (PDU) is that which passes between peer entities. Thus, a PDU on its way down through the layers from one entity to its peer entity, is regarded as a Service Data Unit (SDU) by the layer below (N–1).

Two peer entities can exchange data, with or without a prior connection, at any layer of the hierarchy. A connection is established between two (N) entities by identifying an (N–1) Service Access Point (N–1 SAP) for each of them.

The Service Access Points are uniquely identified by their addresses. These addresses are defined using the following rule:

(N) SAP address = (N) Selector + (N–1) SAP address

The lowest address is the level 3 address, usually called Network SAP address (NSAP address or NSAP). This address identifies a single system within the world of interconnected open systems. The selectors are used to identify services (like file transfer, mailing) in an open system.

Network Interconnections

When Open Systems are not directly connected through the physical layer, they can be interconnected by using an intermediate system as a relay.

The principle of using an intermediate system as a relay, passing the data of other open systems is shown in Figure 4. The same rules for communication between peer entities apply here as for the management of functions and protocols.

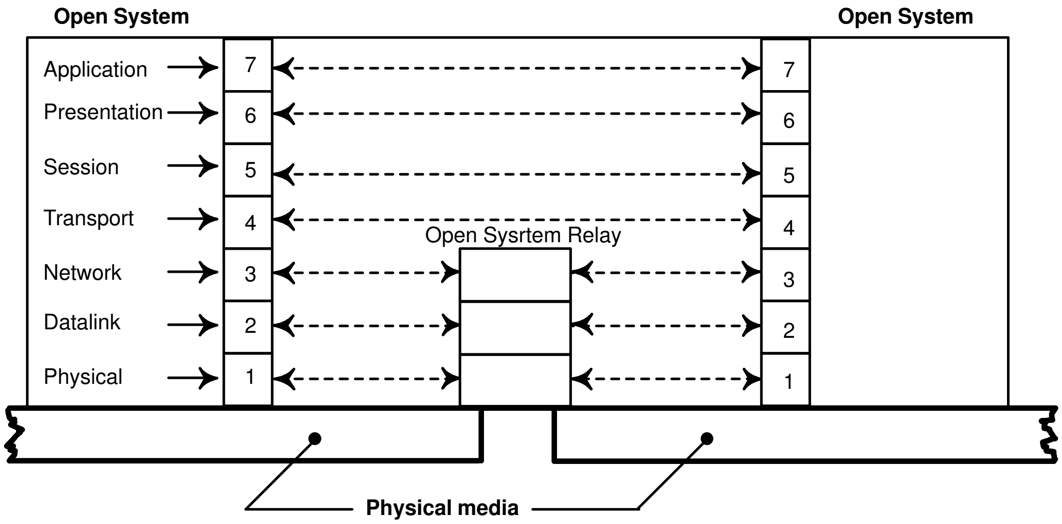


Figure 4. Relaying Principle.

Although the OSI Model specifies only Level 3 relays, OSI Standards define others, such as transport relays (Level 4).

Chapter 2. OSI Stack Characteristics

OSI Stack Characteristics Overview

This chapter is dedicated to administrators or developers requiring detailed information about services, protocols and profiles supported by the OSI Stack.

You can find more information in:

Connectivity

The following communications interface devices are available:

- Ethernet
- Token Ring
- Fiber Distributed Data Interface (FDDI)
- X.25

Details of the physical devices managed by the OSI framework are provided in the *Adapters, Devices, and Cable Information*, see Bibliography, on page B-1 .

Device Common Characteristics

The following characteristics are common to all these devices:

- No specific code is loaded in the board for OSI services. This does not prevent microcode from being loaded, but these microcodes are not part of the OSI Stack product.
- The devices may be shared by the different communications frameworks (OSI, TCP/IP, SNA, and so on).

Additional information is provided in the Software Release Bulletin (SRB) file.

Adapter Location Codes

Communications adapters and lines are identified by a unique 4 to 10 digit code which defines the bus used, the slot number, line and interface used for adapters with several lines.

Example

Here are some examples of location codes.

The location code for the standard Ethernet adapter is as follows:

00–00–0B (where B is a slot number, depending on the CPU board)

Location codes for the Ethernet, LAN, Token Ring and FDDI adapters are as follows:

00–0B (where B is a slot number)

00–1B (where B is a slot number for an adapter connected to the optional MCA bus).

The location code for the HiSpeed WAN Comm. Adapter is as follows:

00–0B–01–0C (where B is the slot number and C the line number).

OSI Stack Functional Description

Here is an overview of the protocol providers which are part of the product. For each provider, the corresponding protocol and service (when available) standard definition is referenced, and then follows a brief description of the supported options and those which are not implemented.

The functional architecture is featured in Figure 5.

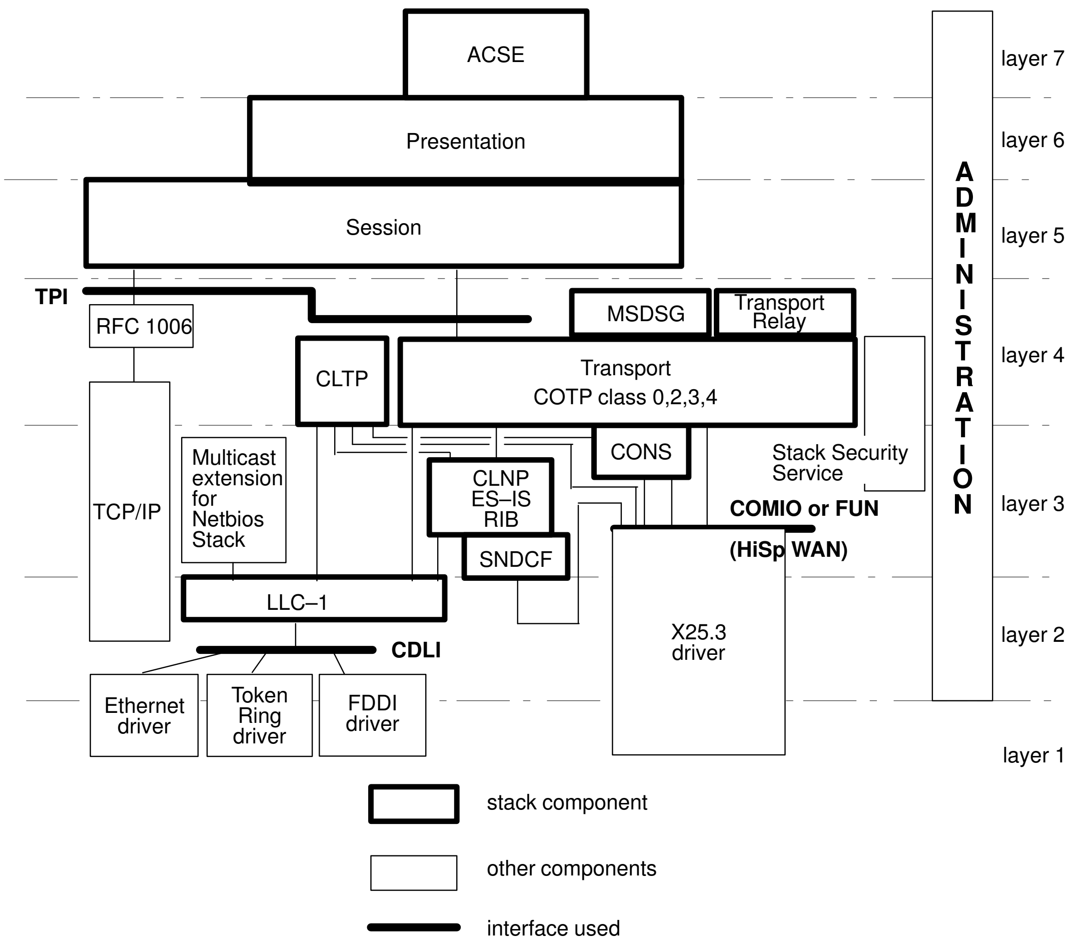


Figure 5. OSI Stack Characteristics – Functional Architecture.

The table summarizes the main standards which are relevant to the implementation of the OSI Stack.

Upper layers	
layer 7	ACSE ISO 8649, 88, ISO 8650, 88
layer 6	Presentation ISO 8823, 88, ISO 8822, 88
layer 5	Session ISO 8327, 87, ISO 8327 DAD2, 88, ISO 8327 DAM3, 91, ISO 8326, 87, ISO 8326 DAM4, 91
Lower layers	
layer 4	TPI, 92 OSI native transport ISO 8073, 88, ISO 8072, 86 ISO 8073/AD1, ISO 8602 MAP/TOP Specification of Netbios interface and name service support by lower layer OSI protocols.
layer 3 (LAN)	ConnectionLess network Protocol, SNDCF ISO 8473, 88, ISO 8348/AD1, ISO 8348/AD2, ISO 8348 ES–IS ISO 9542, 88 MAP/TOP Specification of NetBIOS interface and name service support by lower layer OSI protocols.
layer 2 (LAN)	LLC–1 ISO 8802–2, 89

Association Control Service Element (ACSE)

ACSE provider fully conforms to ISO 8650, 88, plus ISO 8650 TC1 (concerning AE Titles). The service is as defined in ISO 8649, 88. It includes:

- full support of AE Titles form 1 and form 2,
- X410–84 mode available,
- full support of ”rules of extensibility”,
- kernel and authentication functional units supported (see ISO 8650 TC1).

The ACSE provider does not introduce limitations on services provided by the presentation and session providers.

The coding/decoding of the user data transmitted during the association establishment is an ACSE Service–user matter.

Connection Oriented Presentation Service Provider

The Connection Oriented Presentation Provider is fully conformant to ISO 8823, 88 and provides the service defined in ISO 8822, 88. Amendments 2 and 5 are implemented.

The presentation provider supports all the session functional units supported by the session provider. For the presentation specific functional units the kernel FU is supported. The presentation provider supports an unlimited number of contexts per connection, including a default context.

Both normal mode and X.410–1984 mode are supported. Rules of extensibility are also supported.

The size of the user data is limited by the maximum SSDU size. The coding/decoding of the user data transmitted is a Presentation Service–user matter. The COPP only checks the abstract syntax used according to the syntaxes defined in the Defined Context Set (DCS).

Connection Oriented Session Service Provider

The COSS and COSP are as defined in ISO 8326, 87 and ISO 8327, 87, plus ISO 8327 DAD2, 88, ISO 8326 DAD2, 88 for support of unlimited user data, and ISO 8327 DAM3, 91, ISO 8326 DAM4, 91 for Additional Synchronization Functionality (data separation). Defect reports up to 8326/48 and 8327/26 are implemented.

All the Functional Units below are supported:

- Kernel
- Half duplex
- Duplex
- Negotiated Release
- Minor Synchronize
- Major Synchronize
- Resynchronize
- Activity Management
- Exceptions
- Typed data
- Capability Data
- Expedited Data
- Data Separation.

The reuse of transport connections, the use of transport expedited data (when available), and local flow control with the COTP are implemented.

The session provider locally manages the following Quality Of Service (QOS) parameters:

- session connection establishment and release delay,
- session connection priority,
- optimized dialog transfer (to allow concatenation),
- extended flow control.

The session connection priority is ensured by requiring the equivalent QOS to the COTP. The QOS parameters connection establishment and release delay are defined for the whole session provider.

The session provider support the session versions 1 and 2, see Configuration & Limits, on page 4-1. The implementation limits the maximum size of data (even for version 2 which introduces unlimited data for most of the session services).

The COSP is able to use either the native COTS or the COTS on the TCP (using the RFC 1006 implemented in the NetShare product).

Multi–system Distributed System Gateway (MSDSG)

The MSDSG relay is an active transport relay: it performs relaying between CLNS and CONS including class conversion. From the addressing point of view, it behaves like a network relay (i.e. relaying is done according to the called NSAP): its is mandatory to use NSAP addressing of both sides (full CLNP on LAN, CONS on X25). From the protocol point of view, it appears as a single transport connection to the transport user, even if one transport connection is established on each side of the relay. Most of the transport parameters are negotiated end–to–end (extended format, checksum selection, and expedited data, ...). Some parameters may have different values on each side of the transport relay (transport class, TPDU size, ...).

Transport Relay

This relay is a non–standard implementation of transport relays. The transport relay is a TSDU relay. It performs relaying between two transport connections regardless of the profile, network and options used on both sides. The parameters used on the outgoing connection can be configured for each remote network address.

The use of this relay is not transparent for an application.

Connection Oriented Transport Service Provider

The native ISO Transport Service is the Connection Oriented Transport Service (COTS) as defined in ISO 8072, 86. The transport provider is as defined in ISO 8073, (1988).

The transport provider may use either ISO addressing (i.e. NSAPs) over Connection Oriented Network service (CONS) or ConnectionLess Network Service (CLNS), or non ISO addressing (i.e. SNPAs) over LLC–1 services (null CLNP profiles) or over X25.3.

Transport class 0, 2, 3 and 4 are supported over connection oriented service (X25), and only class 4 is supported over connection less service (i.e. over LAN or over X25 through SNDCF). Negotiation is done according to ISO 8073, (1988).

The network service selection is only based on addressing considerations. It does not take into account the QOS of the underlaying service.

The QOS parameter ”connection priority” is negotiated with the remote transport entity at the connection establishment, and is mapped onto a network QOS (parameter priority for CLNS and CONS). Four levels of priority are supported. Data processing is done according to the connection priority allowing higher throughput on connections which are established with higher priority.

The TPDU size is negotiated. The implementation can support all TPDU sizes which are power of two in the range from 128 to 8192. Segmenting is implemented to take into account the NSDU size used by the sub–network. Reassembling and separation are implemented, but concatenation is not available.

On a connection oriented network the transport provider is able to reuse already opened and suitable network connections (same remote node, same priority, allowed multiplexing, same X25 facilities and call user data). On the other hand, the splitting and recombining functions (which allows the use of several network connections for a single transport connection) is not implemented.

ConnectionLess Transport Service Provider

The ConnectionLess Transport Service Provider (CLTP) protocol offers the services defined in ISO 8072/AD1, (1987):

- capability to send and receive UNIDATA TPDU (UN TPDU),
- support of CLNS, inactive CLNS, CONS and X25.3 network services,
- priority management,
- optional use of checksum, capability to send X.25 CUD and facilities,
- use of the configured profiles to supply values not specified by the transport user.

The transport provider may use either ISO addressing (i.e. NSAPs) over Connection Oriented Network service (CONS) or ConnectionLess Network Service (CLNS), or non ISO addressing (i.e. SNPAs) over LLC–1 services (null CLNP profiles) or over X25.3.

The implementation provides local flow control mechanisms to be able to stop applications which send too much data, or to discard incoming UN TPDU if the receive queues are full.

Connection Oriented Network Service & X25.3 Providers

The rules defined in ISO 8878, 87 are used by the CONS component. Only SVCs are supported. CONS service is available on top of X25–84 and 88.

There is always a one to one mapping of address extension facility and the NSAP addresses.

The interface to the connection oriented network ISO 8348 has been extended to allow the network service user to specify the X25.3 facility he wants to use and to be notified of the facilities specified in the X25.3 INCOMING CALL packets.

The protocol providers which provide the X25.3 service are described in ISO 8208.

The OSI Stack does not introduce limitations in the services offered by X25: use of Virtual Call and Permanent Virtual Circuit modes on public or private packet switched or circuit switched networks is supported. All X25–80, 84 and 88 services may be used.

The X25.3 service user may specify facilities for the CALL packet, or be notified of the facilities defined in the INCOMING CALL packet.

The X25.3 service provider is able to force (i.e. to add) facilities in the CALL and CALL CONFIRM packets. The aim is to solve interoperability problems between BULL applications (which do not manage X25.3 facilities) and other vendors.

The X25.3 service user may specify or receive user data during the establishment of a connection.

The X25.3 service provider accepts a maximum size of 4096 bytes for the user data of the DATA packet.

ConnectionLess Network Service Provider (CLNP)

The CLNP complies with ISO 8473, 88, and provides the services defined in ISO 8348/AD1. It may operate over LAN subnetwork (Ethernet, Token Ring, FDDI) using LLC–1 services, or over X25 subnetwork, using SNDCF. Both full and inactive subsets are supported.

The error detection using checksum is performed (its a mandatory function defined in ISO 8473, 88). The segmentation and reassembling function are implemented. It is done according to the local size of subnetwork service data unit (SN–SDU) used (fixed size on Ethernet Token Ring and FDDI, variable size on X25). When routing from a subnetwork to another, total reassembling is performed.

The partial route recording is available.

The CLNP does not manage nor take care of QOS information. When it is not possible to use the selected route (because of a network failure for example), the CLNP does not try to use another route. The single QOS parameter managed by the CLNP entity is the priority parameter: the corresponding QOS is transmitted to the sub–network service used.

The CLNP entity accepts the NSAPs according to ISO 8348/AD2, but any NSAP are supported: names are handled as octets string and no syntax control is performed (except length).

The CLNP queries the Routing Information Base and activates the forwarding process (see Routing Framework, on page 2-8).

The CLNP entity also queries the RIB for incoming messages which must be routed. It is able to route on any couple of supported subnetworks: LAN/LAN, LAN/WAN and WAN/WAN.

Subnetwork Dependent Convergence Function

This function offers the ConnectionLess service assumed by CLNP over connection oriented networks at the packet level protocol over the X25.3 service. The service is provided over packet switching sub–networks and circuit switched sub–networks. Permanent Virtual Circuits (PVCs) are not supported by this SNDCF).

To transmit a PDU, the SNDCF requests the opening of a Virtual Circuit (VC) between the local and remote SNPAs, if such a VC is not yet available. If the VC can not be established, the PDU is lost. The VC is closed after an idle time.

The function does not take care of the QOS requirements because the CLNP never specifies QOS requirements.

Routing Framework

The routing framework conforms to ISO TR 9575, 89. It is based on static routing over LAN and WAN (both connection less and connection oriented mode) and dynamic routing using ES–IS protocols on LANs.

The activation of the ES–IS protocol is configurable, and when the protocol is active, a station is either ES or IS or both. The RD PDUs are sent by IS and processed by ES. The configuration timer of ES is recomputed on the receipt of a lower suggested ES configuration timer information.

Local Area Network Protocol Provider

The LAN protocol provider conforms to the global specification ISO 8802–1, (1989). It contains a Logical Link Control (LLC) sublayer which provides Un–acknowledged Connectionless mode service (LLC type 1, as defined in ISO 8802–2, 89), and uses the services of several Medium Access Control (MAC) sublayers.

These services are offered by the Ethernet, Token Ring and FDDI device drivers through the CDLI interface. It is assumed that they conforms to ISO 8802–3 (for Ethernet), ISO 8802–5 (Token Ring), and ISO 9314–2 (for FDDI).

The LLC is able to answer to XID and TEST commands, but it never originates such commands.

Network Management

Most of the stack components are managed objects, but no application programming interface is provided to access to these information.

The product ”CMIP Agent for OSI Layers” allows the following management:

- Application layer,
- Presentation layer,
- Session layer,
- Transport layer (Connection Oriented profile),

- Network layer (ConnectionLess profile),
- X.25 level 3.

See ”CMIP Agent for OSI Layers– Installation, Administration & User’s Guide, Bibliography on page B-1.

Security Service

The OSI Stack contains a security service which filters incoming transport connections (COTP), transport datagrams (CLTP) and NPDU relayed to remote systems (CLNP relay function). Filtering is done according to calling and called network addresses (NSAP, SNPA or PVC), and also according to the called TSEL for the transport protocols.

Multicast Feature for Netbios

The support of Netbios services on top of OSI transport services (with both connection oriented and connectionless modes) uses some extensions of the of the transport service which are not defined in OSI standards. The aim is to allow the multicasting of ConnectionLess Transport PDUs (UN TPDUs). A new service, based on full CLNP profile, but without a routing function and with restricted error reporting using a configurable LSAP (default is 0xEC), is used.

CLNP manages a multicast NSAP, which is not handled as a local NSAP. This NSAP can be used only through CLTP. When it is used (always as a destination NSAP or as a bind address), CLNP operates over LLC–1 using the Netbios LSAP and a standardized multicast MAC address. When receiving an NPDU with this NSAP as called address, CLNP does not attempt to route the NPDU, but delivers it to CLTP.

The multicast feature cannot operate on several networks. Only Ethernet and Token Ring are supported.

Profiles

This paragraph describes communications profiles for OSI layers 1 to 4. A brief description of each profile is provided.

References to standardized profiles are listed in OSI Telecommunications Profiles, on page A-1.

The following profiles are supported:

- End systems
  - LAN profiles
  - X25 profiles
- Intermediate systems:
  - LAN/LAN profiles
  - LAN/X25 profiles
  - X25/X25 profiles.

End Systems

The entry "Names:" after each Profile table refers to the names usually used to identify the corresponding profile.

LAN Profiles

The protocols used by these profiles are described in the next tables.

transport	ISO 8073/ADD.2 (class 4)
network	ISO 8473
data link	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Names: full CLNP, Internet, 3, CL.

transport	ISO 8073/ADD.2 (class 4)
network	ISO 8473 inactive subset, LSAP 0xFE
data link	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Names: null CLNP, Ethernet, RLE, 4, LAN.

transport	ISO 8073/ADD.2 (class 4)
network	ISO 8473 inactive, any LSAP
data link	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Names: DSA, configurable null CLNP, 2, LANC.

These profiles offer transport class 4 over Ethernet, Token Ring or FDDI networks. The choice of the physical medium is fully transparent from the transport user point of view.

When operating with full CLNP profile, the transport user must provide calling and called NSAPs. The routing framework is used to route the PDUs. Use of full CLNP profiles is mandatory for CLNP or MSDSG relaying.

With inactive subset, the transport user must provide the calling and called physical addresses.

The profile with an inactive subset and any LSAP can be used as a DSA profile when the LSAP used is configured to 0x20.

The following table provides input to OSI Telecommunications Profiles, on page A-1.

	full CLNP	Null CLNP inactive subset ISO LSAP 0xFE	Null CLNP inactive subset LSAP 0x20	Null CLNP inactive subset other LSAP
Ethernet	LAN/2	LAN/1	SID	none
Token Ring	LAN/4	LAN/3	none	none
FDDI	LAN/6	none	none	none

Note: 'none' means that no standardized profiles have been identified for this configuration.

Similar profiles are available for the connectionless transport protocol.

transport	ISO 8073/ADD.1, connectionLess
network	ISO 8473
data link	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Names: full CLNP, Internet, 3, CL.

transport	ISO 8073/ADD.1, connectionLess
network	ISO 8473 inactive subset, LSAP 0xFE
data link	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Names: null CLNP, Ethernet, RLE, 4, LAN.

transport	ISO 8073/ADD.1, connectionLess
network	ISO 8473 inactive, any LSAP
data link	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Names: DSA, configurable null CLNP, 2, LANC.

X25 Profiles

Several profiles are available using either connection oriented service over X25.3 (with or without CONS), or CLNS service.

The first profile is described in the next figure. It uses one of the supported transport class over X25.3. The transport user must provide the calling and called X25 addresses, or PVC identification. Every X25 versions are supported.

Support of X21 bis and X21 switched lines or leased lines depends on the communications adapter used.

transport	ISO 8073 (class 0,2,3,4)
network	ISO 8208 (X25.3)
data link	ISO 7776 (LAP–B)
physical	X21 or X21bis

Names: X.25, 1 or 6, SVC or PVC.

This profile is referenced as X25/1 in OSI Telecommunications Profiles, on page A-1.

The second profile is similar but provide CONS service on top of X25.3 (as defined in ISO 8878, 87). Only SVCs are supported on X25–84 or 88 are supported.

This profile is mandatory for MSDSG relaying.

transport	ISO 8073 (class 0,2,3,4)
network	ISO 8878 (CONS) ISO 8208 (X25.3)
data link	ISO 7776 (LAP–B)
physical	X21 or X21bis

Names: CONS, SPEE, 5, CLCO.

This profile is referenced as X25/5 in OSI Telecommunications Profiles, on page A-1.

The last X25 profile provides CLNS on top of X25.3 (using SNDCF). Only transport class 4 is available (because of CLNS). SVCs are supported on all X25 versions.

transport	ISO 8073/ADD 2 (class 4)
network	ISO 8473 (including SNDCF) ISO 8208 (X25.3)
data link	ISO 7776 (LAP–B)
physical	X21 or X21bis

Names: Full CLNP, Internet, 3, CL.

This profile is referenced as X25/2 in OSI Telecommunications Profiles, on page A-1.

Similar profiles are available for the connectionless transport protocol.

transport	ISO 8073/ADD.1, ConnectionLess
network	ISO 8208 (X25.3)
data link	ISO 7776 (LAP–B)
physical	X21 or X21bis

transport	ISO 8073/ADD.1, ConnectionLess
network	ISO 8878 (CONS) ISO 8208 (X25.3)
data link	ISO 7776 (LAP–B)
physical	X21 or X21bis

transport	ISO 8073/ADD.1, ConnectionLess
network	ISO 8473 (including SNDCF) ISO 8208 (X25.3)
data link	ISO 7776 (LAP–B)
physical	X21 or X21bis

Intermediate Systems

LAN/LAN

The first profile uses the CLNP relay requiring CLNP protocol on both side of the relay.

network	ISO 8473	
data link	ISO 8802.2 type 1	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO8802.5 (token ring) or ISO 9314 (FDDI)	ISO 8802.3 (CSMA/CD) or ISO8802.5 (token ring) or ISO 9314 (FDDI)

This profile is referenced in OSI Telecommunications Profiles, on page A-1 by the following profiles:

	Ethernet	other
Ethernet	REL/2	none
Token Ring	none	none
FDDI	REL/ter	none

Note: 'none' means that no standardized profiles have been identified for this configuration.

The second profile uses the MSDSG relay or transport relay.

	MSDSG or Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473	ISO 8473
data link	ISO 8802.2 type 1	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

The last set of profiles uses the transport relay and is described in Tables a. and b.

	Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473	ISO 8473 inactive subset
data link	ISO 8802.2 type 1	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Table a.

	Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473 inactive subset	ISO 8473 inactive subset
data link	ISO 8802.2 type 1	ISO 8802.2 type 1
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)

Table b.

LAN/X25

The first profile uses the CLNP relay function and CLNP protocol on both LAN and X25.

Referenced as REL/2 bis.

network	ISO 8473 / ISO 8208	
data link	ISO 8802.2 type 1	ISO 7776
physical	ISO 8802.3 (CSMA/CD) or ISO8802.5 (token ring) or ISO 9314 (FDDI)	X21

The second set of profiles use the MSDSG relay or transport relay and is described in Tables a. and b.

Referenced as REL/1 bis.

	MSDSG or Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473	ISO 8473 / ISO 8208
data link	ISO 8802.2 type 1	ISO 7776
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	X21

Only SVCs are supported.

MSDSG relay does not support PVCs.

Table a.

	MSDSG or Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473	ISO 8473 / ISO 8208
data link	ISO 8802.2 type 1	ISO 7776
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	X21

Only SVCs are supported.

MSDSG relay does not support PVCs.

Table b.

The third set of profiles is supported only by the transport relay and is described in Tables a., b. and c.

	Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473	ISO 8878
data link	ISO 8802	ISO 7776
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	X21

Table a.

	Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473	ISO 8878 / ISO 8208 **Note:** PVCs not supported.
data link	ISO 8802	ISO 7776
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	X21

PVCs not supported.

Table b.

	Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473 inactive	ISO 8208 **Note:** PVCs and X25–80 not supported.
data link	ISO 8802.2 type 1	ISO 7776
physical	ISO 8802.3 (CSMA/CD) or ISO 8802.5 (TokenRing) or ISO 9314 (FDDI)	X21

PVCs and X25–80 not supported.

Table c.

X25/X25

The first profile uses the CLNP relay function.

network	ISO 8473 / 8208	
data link	ISO 7776	ISO 7776
physical	X21	X21

PVCs not supported.

The second set of profiles requires NSAP addressing and is supported by both the MSDSG relay and transport relay.

	MSDSG or Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473 / ISO 8208	ISO 8473 / ISO 8208
data link	ISO 7776	ISO 7776
physical	X21	X21

MSDSG supports only relaying from CLNP to CONS and not from CONS to CLNP.

No PVCs. X25 – 80 not supported.

The final set of profiles is supported only by the transport relay.

	Transport relay	
transport	ISO 8073/ADD.2 (class 4)	ISO 8073/ADD.2 (class 4)
network	ISO 8473 / ISO 8208	ISO 8473 / ISO 8208
data link	ISO 7776	ISO 7776
physical	X21	X21

PVCs not supported.

Certification and Conformance

The OSI Stack aims to comply to ISO standards, International Standardized Profiles (ISP), Government profiles (US or UK GOSIP), and European profiles (ENV) within the limits of their compatibility.

Interoperability

OSI to OSI Links

The OSI Stack is implemented according to OSI standards and international profiles. This is a guarantee for interconnection and interoperability with equivalent OSI products. The flexibility of the stack configuration also wide the scope of inter–operable systems.

Interoperability is ensured with OSI Stacks and OSI applications delivered by BULL on UNIX platforms.

Interoperability with GCOS8, GCOS7, CGOS6 and GCOS4 platforms is also possible through applications like Message Handling (X400), file transfer (FTAM), transactional processing (CPI–C OSI, BOS/TP), PAD and SQL database access (SQL*NET), depending on the application actually supported by the considered platform.

OSI to DSA Links

The OSI Stack provides support for OSI–DSA communications links which allow GCOS connectivity using specific applications like file transfer (UFT), terminal emulation (OTM), transactional processing (BOS/TP). Such communication links require to have a session protocol conversion between OSI session and GCOS session. This conversion is usually done in a GCOS front end.

Other Links

No general rules apply to other types of links. Interoperability must be studied on a per–case basis.

Security & Data Integrity

No security protocols are provided.

A security function allows information to be filtered according to addressing parameters.

Data integrity is ensured by implementation (user data transmitted is not modified by the stack), but there is no guarantee that data is transmitted: all ISO protocols support a connection abort service which allows a connection to be terminated without any guarantee that data is delivered.

Chapter 3. Addressing

The aim of this chapter is to provide a consistent view of the addressing within the OSI Stack and to explain how the reference model has been implemented. It is dedicated to administrators and developers who need a fine knowledge of stack addressing implementation.

Addressing Overview

This section provides a detailed view of addressing from within the OSI Stack. Wildcarding – a powerful mechanism used extensively in transport and upper layers – is introduced. A list of addressing parameters is provided, with form and explanations. The routing framework and Routing Information Base are described. The dispatching of incoming and outgoing calls is also explained, together with the relaying mechanism. You can find more information in:

- Creation of Service Access Points, on page 3-2.
- Wildcard Addressing Concepts, on page 3-3.
- Addressing Parameters, on page 3-6.
- Routing Framework and RIB, on page 3-10.
- Outgoing Calls, on page 3-13.
- Incoming Calls, on page 3-15.

Creation of Service Access Points

The OSI model is structured in layers, which exchange information using a request/indication/response/confirmation scheme through Service Access Points. These service access points are identified by an address which is unique in an addressing domain. Applications initiating a connection must specify the calling and called addresses (i.e. list of selectors and a NSAP). The called application is retrieved from the called address. But the reference model never deals with SAP creation or deletion.

The real world is a little bit different: SAPs are usually associated with resources, queues, ... which must be allocated and released. The rules attached to SAP creation are summarized in the next table.

ACSE/Presentation ... transport	connection originator: dynamic creation of SAPs, SAP addresses are used for calling address	connection recipient: dynamic creation of SAPs, matching called address and SAPs are used for dispatching
network	static configuration of NSAPs	
data link	default configuration of LSAPs (on LAN)	
physical	static configuration of physical addresses	

SAPs presentation to transport are created dynamically through explicit bind operations. The SAPs have different properties and different limits applied because the bind is issued by a connection originator or a connection recipient. For a connection originator, the SAPs created are used as calling addresses for subsequent calls. Some APIs do not show a bind service, but only a connect service. In that case, the bind is automatically issued with the calling address. A connection recipient receives all the incoming connections whose called address ”matches” the bind address. If there is no recipient, the incoming connections will be rejected by the providers.

For relaying purposes, it is necessary to receive incoming calls or PDUs even if there are no recipient applications. The same rule applies when dynamic routing protocols are used: it is necessary to multicast addresses which may exist on a station even if they are not used at the present time. These two reasons lead to a static configuration (i.e. usually by an administrator) of the network, data link and physical SAPs.

Wildcard Addressing Concepts

The mechanisms defined in Creation of Service Access Points are very compulsive: the originator application (also called client) must fully specify a local calling address and a remote called address. The recipient application must fully specify the expected called address.

Wildcard mechanisms have been implemented to avoid this full specification for both connection originator and connection recipient.

Wildcarding for Connection Originator

In the bind, if no TSAP is provided, a TSAP (random TSEL, wildcard NSAP) and the wildcard LSAP are generated (only when accessing the TPI interface using XTI API). It is also possible to use a wildcard NSAP or LSAP for the bind. At the establishment of a connection, the wildcard NSAP triggers the use of the first NSAP of the local NSAPs list for the calling NSAP and the use of LSAP 0xfe.

Bind parameter	**Provided by Stack for calling address**
no TSAP	random TSEL, wildcard NSAP
wildcard TSEL	random TSEL
wildcard NSAP	one of the local NSAPs
no NSAP	not supported
wildcard LSAP	0xfe, previously defined*
no LSAP	0xfe, previously defined*

* Configured default LSAP. Value can be changed. See How to Define NULL CLNP Profile Network LSAPs, on page 9-108.

Note: There is no similar mechanism for upper layer selectors.

Wildcarding for Connection Recipient

Wildcarding for connection recipient is much more complex. A wildcard address is a sort of garbage–can which receives the incoming calls (or PDUs) which are not received by other applications using SAPs which do not contain wildcard selectors in the bind address.

This behavior is featured in Figure 6.

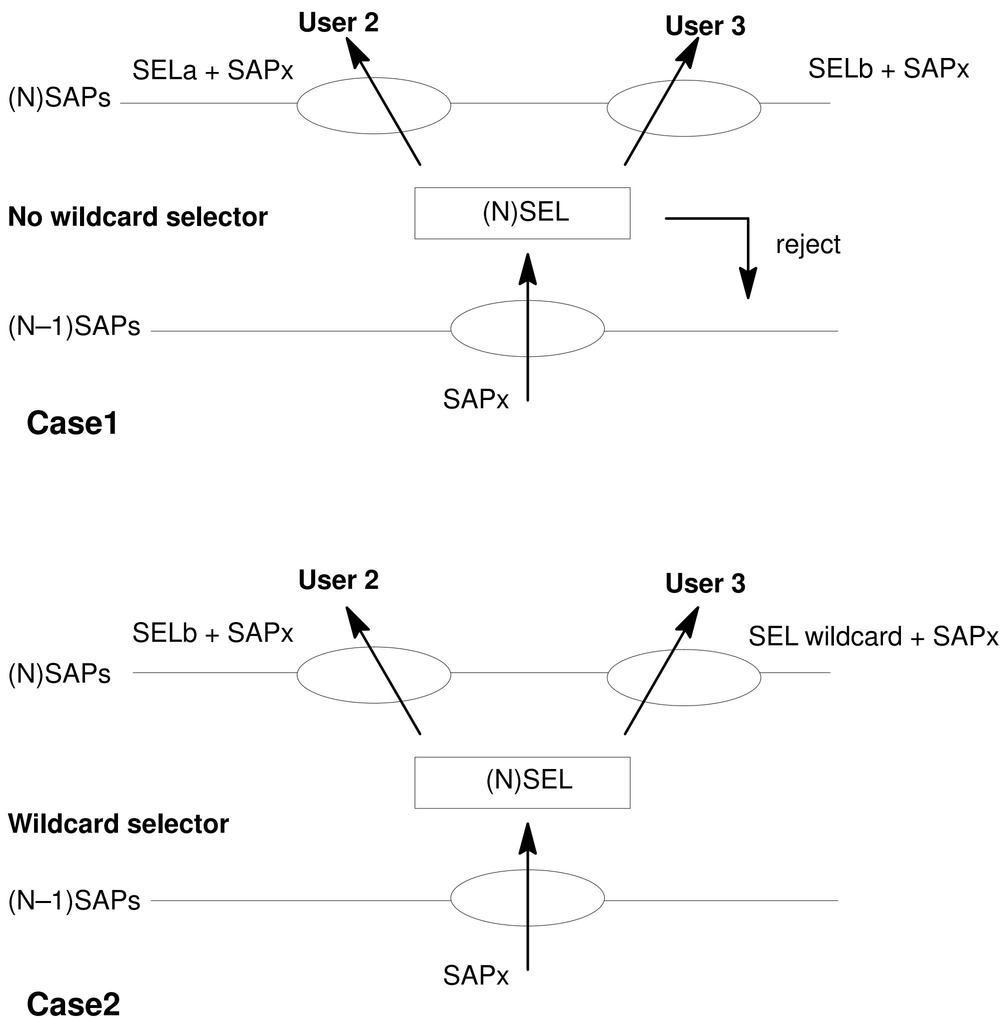


Figure 6. Wildcarding

In case 1, an incoming connection is rejected if the called selector is neither SELa or SELb. But in case 2, all the incoming connections, which do not have SELa as called selector, are delivered to the user listening with the wildcard selector.

This mechanism lead to conflicts when wildcard selectors are supported by several protocol providers as shown in Figure 7.

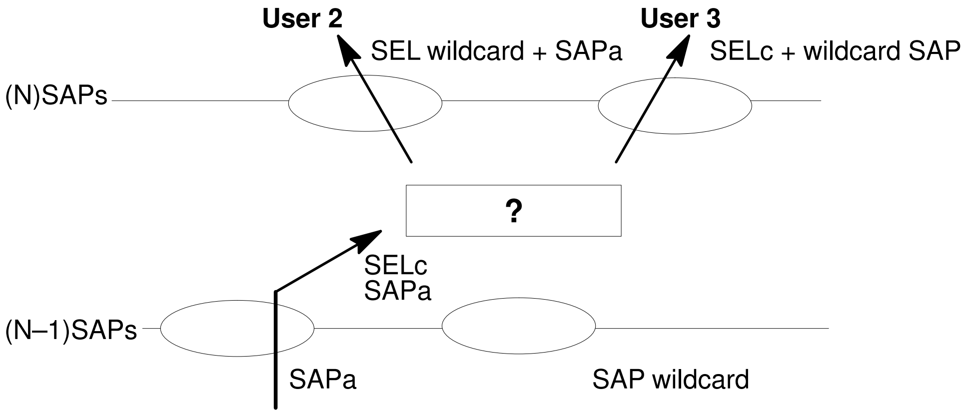


Figure 7. Conflicts Due to Wildcarding

The incoming connection received on (N–1) SAPa with a called (N) selector SELc can be delivered either to user 2 listening with a wildcard (N) selector and (N–1) SAPa or to user 3 listening with (N) SELc and a wildcard (N–1) SAP.

Such problems are solved by priority rules, which define the priority level of each selector: the incoming calls are delivered to the first user (according to the priority rule) who matches the called address.

Addressing Parameters

Characteristics

The following table summarizes all the addressing parameters used in the Stack and their characteristics.

Name	Definition	Coding	Specific Values	Comment
AE Title	dynamic	ASN.1 (form 1 & 2)		Up to 2048 different AE Titles. (1 per connection).
PSEL	dynamic	0..16 byte hexa	4 bytes null (wildcard PSEL)	Up to 2048 different PSELs (1 per connection).
SSEL	dynamic	0..16 bytes hexa	4 bytes null (wildcard SSEL)	Up to 2048 different SSAPs (1 per connection).
TSEL	dynamic, except for TPAD–HPAD*	0..32 bytes hexa	8 bytes null (wildcard TSEL) 1 predefined TSAP (TSEL=0x40.01.50.41.44.33, NSAP = any)	
NSAP	configured by administrator	1..20 bytes Binary Predefined encoding.	null length NSAP (wildcard NSAP)	Up to 32 different NSAPs (shared by CLNS & CONS). First NSAP is used as NET value. See NSAP Syntax. Plus 1 NSAP reserved, see Multicast Feature for NetBIOS.
LSAP	predefined and / or configured by administrator	7 bits hexa	0xfe: OSI services 0x20: DSA services 0xFF: wildcard LSAP	FF is only an interface convention.
SNPA (LAN)	configured by selecting communications adapters	6 bytes hexa		1 per LAN access point.
SPI	predefined, configuration	1..4 bytes hexa	18 predefined values (including no value & unknown value) See SPI	See SPI
SNPA (DTE address)	configured by selecting communications adapters	1..15 bytes BCD, coded length in quartets		1 per X25 line.
PVC	configured by administrator	1..8 bytes (ASCII character string)		See PVC

* See special support for TPAD–HPAD.

NSAP Syntax

The NSAP syntax is defined in ISO 8348.

The two main parts of an NSAP address are IDP and DSP, as follows:

Initial Domain Part (IDP): This part supplies the code and syntax used for addressing and information on the global domain to which the system belongs.
The IDP part is composed of two components:

- Authority Format Identifier (AFI) which is used to specify:
  - the IDI format
  - the authority in charge of allocating IDI values
  - the abstract syntax of the DSP.
- Initial Domain Identifier (IDI) which, depending on the AFI allocation value, specifies:
  - the addressing domain in which the DSP values have been allocated
  - the authority in charge of allocating DSP values in this domain.

Domain Specific Part (DSP): This part gives information on the system position in relation to the domain where it belongs.
The NSAP address format is shown in Figure 8.

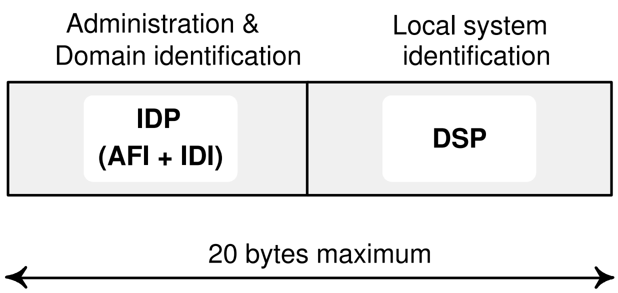


Figure 8. NSAP Format.

ISO 8348 also defines the Preferred Binary Encoding of NSAPs, which is used by CLNP to transmit NSAP addresses. Rules are defined for coding the abstract NSAP syntaxes (AFI, IDI, DSP) as well as for padding characters (which ensure an entire number of bytes for both IDI and DSP parts).

The OSI Stack does not check the NSAP syntax according to ISO 8348; any NSAP is supported. The NSAP syntax is not used to get addressing information except within the session layer. All connections established through RFC 1006 (NetShare) must have calling and called NSAP with a well–defined syntax (including a fixed prefix and a coding or TCP/IP address), see NetShare User's Guide, Bibliography, on page B-1.

CLNP (for partial route recording) and ES–IS (for ISH PDUs) require the definition of NET per station. The first NSAP is used for NET definition.

NSAP Examples:

AFI formats:

```
AFI = 36 Consultative Committee on International Telegraphy &
       Telephony (CCITT) X121 address in decimal code
```

```
AFI = 38 ISO Data Country Code (DCC) address in decimal code
```

```
AFI = 47 ISO  International Code Designator (ICD) address in
       binary code
```

```
AFI = 49 local address in binary code
```

IDI formats:

```
If AFI = CCITT, then IDI = TRANSPAC address.
```

```
If AFI = ISO, then the IDI includes the identification of the
organization in charge of allocation (for example, AFNOR in
France).
```

Syntax of an NSAP address for the **ISO/DCC** authority using the DSP abstract **decimal coding:**

```
AFI: 38

IDI: 250 (AFNOR identification)

DSP: 000006325 (recorded AFNOR organization)
+ 12341234567890123456l2 (part reserved for the organization)

NSAP = 38 250 000006325 1234123456789012345612
```

Syntax of an NSAP address for the **ISO/DCC** authority using the DSP abstract **binary coding:**

```
AFI: 39

IDI: 250 (AFNOR identification)

DSP: F0018b5 (recorded AFNOR organization)

+ 8A5F123456789ABCDE (part reserved for the organization)

NSAP = 39 250 F0018b5 8A5F123456789ABCDE
```

Syntax of an NSAP address for the NSAP **ISO/ICD** authority using the DSP abstract **binary coding**:

```
AFI : 47

IDI : 0005

DSP :  0032 : organization ID
       1234 : subnetwork identifier
       53 18 44 27 : system address in the subnetwork
       01 : NSAP selector

NSAP = 47 00 05 00 32 12 34 53 18 44 27 01
```

Syntax of an NSAP address that is **not administered by a Standards Organization** and using the abstract **binary coding:**

```
AFI : 39

IDI : 250f (padded with "1" at right)

DSP :  000025 : organization name
       0000 : subnetwork name
       138123456fff : system X.121 address (padded with "1" at
       right)

NSAP = 39 250f 000025 0000 138123456fff 00
```

Subsequent Protocol Identifiers

Protocol Identifiers (PI) and Subsequent Protocol Identifiers (SPI) are defined in ISO 9577. These parameters are 1 byte in length and allow protocols transmitted in link layer (2) Service Data Unit (SDU) to be identified.

The OSI Stack implementation extends the SPI definition to parameters of 4 bytes in length. These values are used to dispatch incoming X25.3 call packets.

Applications do not need to provide a SPI value in the X25.3 Call User Data (CUD). The OSI Stack supplies the appropriate values. The management of received CUD is configurable for the purposes of interoperability. See Use of X25 Subnetworks, on page 6-5.

The default SPI configuration is as follows:

- no data, 0x00, 0x03:   transport
- unknown, 0x01, 0x02: X25.3 access method
- 0x81:   connectionless network layer

Local SNPA (DTE Address)

For X25.3 addresses, there is no control to check if the address respects the x.121 specification. The local SNPAs are obtained using the configured communications adapters.

Applications using X25.3 addresses must code the length in quartet and the value in BCD.

Note: This is the only addressing parameter which is not coded in bytes.

Local SNPA (LAN)

The local SNPA addresses are obtained from the configured communications adapter. There is no need to redefine local addresses within the Stack configurator.

The Ethernet, Token Ring and FDDI addresses have the same length (6 bytes). The use of a specific LAN is fully transparent for applications.

Applications using SNPA addresses must code the length in bytes.

PVC

PVCs must be configured by the administrator, who defines for each PVC a name (used by applications to identify the PVC regardless of the circuit number) and the entity which manages the PVC (transport or X25.3 Access Method).

Applications using PVCs must code the length in bytes (name length). When this address is mapped onto a network address (i.e. for applications over the transport layer), it is coded in the called address field.

Routing Framework and Routing Information Base

Overview

The routing framework is used by the network layers as defined in ISO TR 9575 for both CLNS and CONS. It allows the routing of outgoing PDUs and the relaying of incoming PDUs not destined for the local system. The routing information is stored in the Routing Information Base (RIB). The OSI Stack supports both static routing (RIB configured by the administrator) and dynamic routing (RIB updated by the ES–IS protocol).

Figure 9 shows the routing information mechanism.

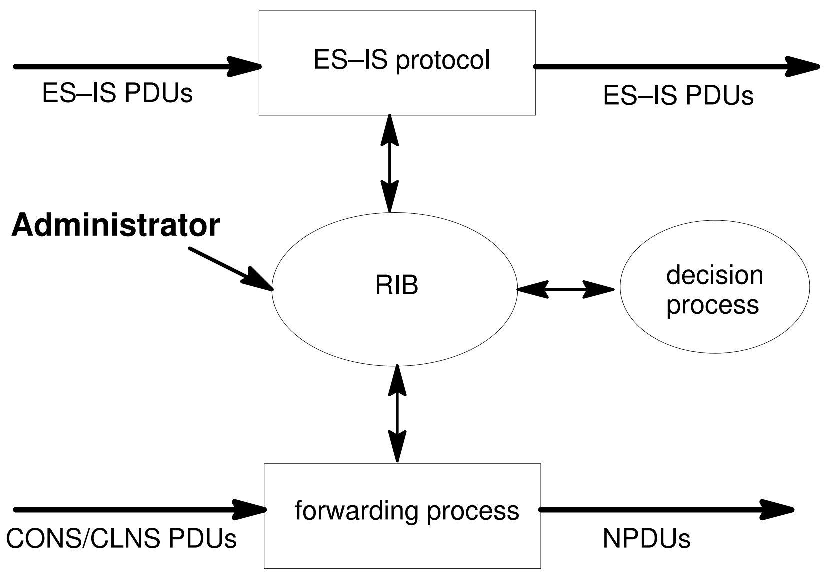


Figure 9. Routing Information Mechanism

Static & Dynamic RIB

The RIB part managed by the ES–IS protocol is called **dynamic RIB**. The part managed by the administrator is called **static RIB**.

Common parameters are available for every RIB entry, regardless of the type of RIB:

- Remote NSAPs
- Local SNPA (subnet for static RIB)
- Remote SNPA (PVC or computation algorithm for static RIB).

Static RIB contains the following specific parameters:

- Mask for remote NSAP (defines sets of remote NSAPs in a single RIB entry)
- Routing method (CLNP/MSDSG)
- Priority
- RIB entry state (Enable/disable)
- PDU lifetime
- CUG
- RIB part.

The RIB part parameters define how the RIB entry is managed.

- STATICF entries are always loaded in stack memory and are always available when a route is searched. The STATICF entries are also called ”fixed entries”.
- STATICV entries are only partially loaded in stack memory, using a cache mechanism between memory and disk. Only the subset of entries loaded in memory are available when a route is searched. The STATICV entries are also called ”variable entries”.

RIB Search

Here is explained how static and dynamic RIBs are used by both CLNS and CONS.

Figure 10 shows the data structure and mechanism used to store routing information.

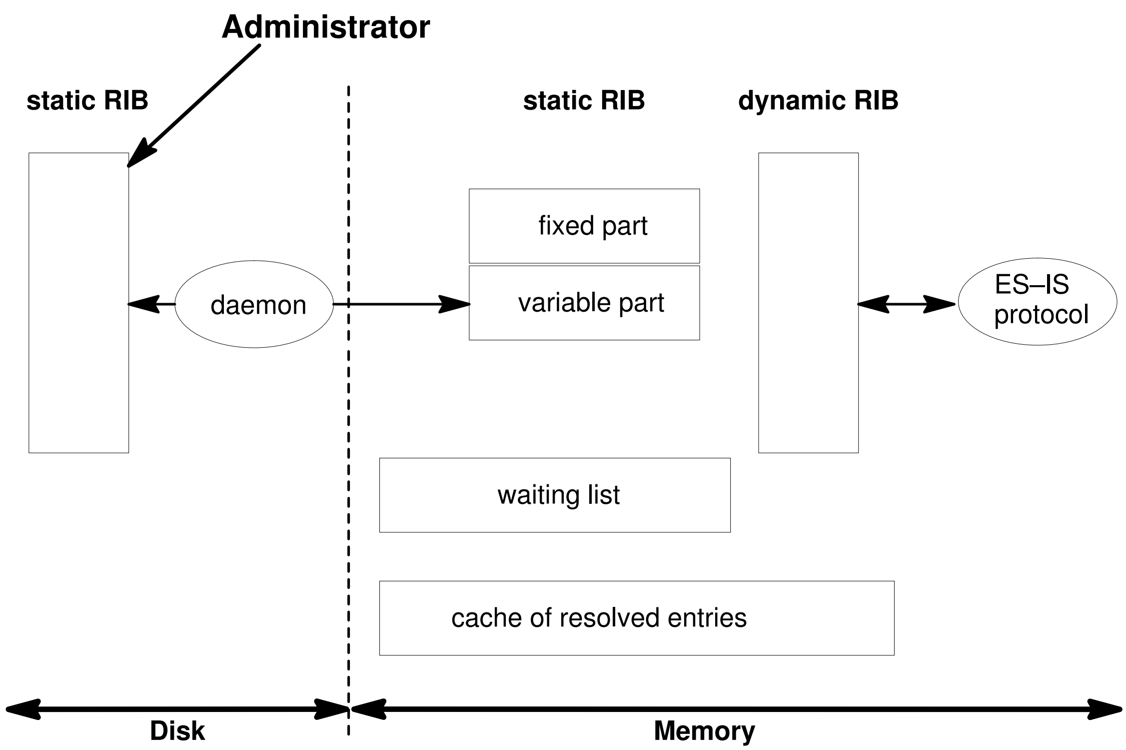


Figure 10. RIB Data Structure Mechanism

RIB Priority

Static RIB has a higher priority than dynamic RIB.

The search algorithm is as follows:

- search in the last resolved routes (small cache with two entries)
- search in the waiting list

if not found

- search in the static RIB

if not found

- search in the dynamic RIB

if not found and ES–IS protocol is enabled

- search for an IS in the dynamic RIB.

The search fails if: nothing is found, no route, no IS, or, if the route is found in the waiting list.

If nothing is found, the remote NSAP is added in the waiting list and the cache mechanism for the variable static RIB is activated.

When the CLNP is notified of a search failure, it multicasts the NPDU to all the ES (if the ES–IS protocol is active) or it discards the NPDU.

Dynamic RIB is managed only on LAN. Routes used for CONP must be defined in static RIB. If CONP is notified of a search failure, it tries to send the NPDU using CLNP. Therefore CONS routes must be configured as fixed routes.

It is possible to have conflicts when using masks to define several routes in the same static RIB entry. These conflicts are partially solved using a priority algorithm taking the user–defined priority and mask selectivity into account.

The user–defined priority is used first. If it does not allow the conflict to be solved, the mask selectivity is then used. (A mask which is not very selective, i.e. with many zeroes, has a low priority). Because of the variable entries, it is always possible to have a route with a high priority which is not loaded in memory. If the overlap between fixed and variable entries cannot be avoided, it is advisable to define a higher priority for the fixed entries (compared with variable entries). When overlapping is unavoidable, the results of the search for these routes cannot be predicted.

Outgoing Calls

Switching of outgoing calls in the session, transport and CLNP layers is shown in Figure 11.

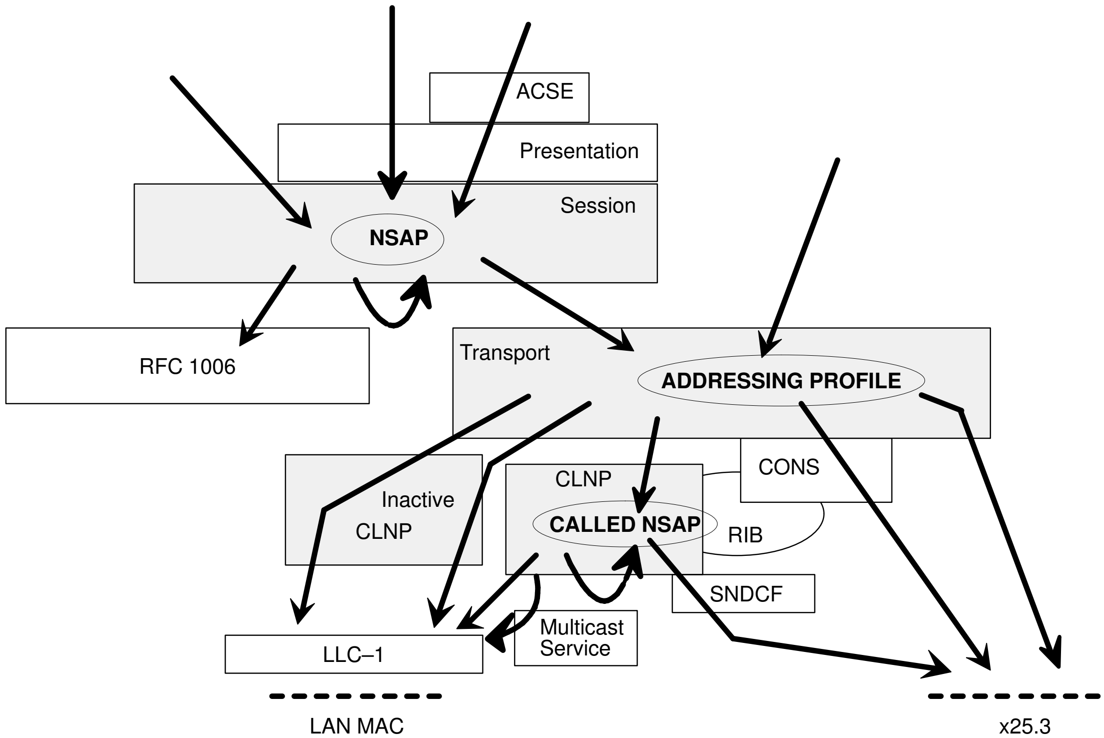


Figure 11. Outgoing Calls Switching.

Lower Layers

Transport users select, at connection establishment time, one of the five supported addressing profiles. When ISO profiles are selected (i.e. NSAP addressing) the RIB solves the addressing problem:

called NSAP → local SNPA + remote SNPA + additional protocol parameters.

(CLNP may activate dynamic routing protocol to get information).

The type of addressing profile is defined on a 'per connection' basis. The coding of this information depends upon the API used (XTI, XAP, ROSE, Session). The use of these extra parameters avoids making a hypothesis on the NSAP syntax (to select a profile according to an addressing scheme) or to have extra configuration mechanisms to link local addresses to addressing profiles.

When using ISO profiles, it is possible to use a wildcard NSAP for the calling address, see Wildcard Addressing Concepts, on page 3-3.

CLNS and CONS do no use the same RIB services, see Routing Framework and RIB, on page 3-10.

When CLNP is used, a loopback may be performed when both calling and called NSAPs are local.

If the called NSAP is equal to the NetBIOS NSAP, the NPDU is sent using a MAC multicast address and NetBIOS LSAP, see page 2-9 and page A-1.

The addressing profile parameter refers to the end system profiles described in OSI Telecommunications Profiles, on page A-1. The correspondence between end system profiles and the addressing profiles is provided in the next table. The way to specify the addressing profile depends on the interface used.

LAN profile (full CLNP)	LAN profile (inactive CLNP, LSAP 0xFE)	LAN profile (inactive CLNP, any LSAP)	X25 profile (without CONS)	X25 profile (with CONS)	X25 profile (CLNS over X25)			
LAN/2 LAN/4 LAN/6	LAN/1 LAN/3	SID	X25/1	X25/5	X25/2	specification for TPI interface	specification for upper layer interfaces	specification for profile configuration
			X (SVC)			null CLNP (WAN SVC)	1	SVC
		X				null CLNP (any LSAP)	2	LANC
X					X	full CLNP	3	CL
	X					null CLNP (ISO LSAP)	4	LAN
X				X		SPEE	5	CLCO
			X (PVC)			null CLNP (WAN PVC)	6	PVC

The TPI interface is available through the XTI API. The addressing profile is defined using three basic values (null CLNP, full CLNP, SPEE), and shades of meaning are introduced by the format of the address parameter (NSAP, LAN address, SVC address, PVC name, LSAP).

Upper Layers

The upper layer interfaces are available through XAP, ROSE and session APIs. The addressing profile is identified by a value in the range from 1 to 6.

The applications must provide the calling upper selectors. There are no wildcard selectors or generation mechanisms as for the transport provider.

When using NSAP addressing, the session layer provides switching between TCP/IP and OSI transport providers according to the NSAP value:

- NSAPs starting with the value 0x54.00.72.87.22.03 (AFI=54, IDI=728722, DSP=03<IP address. 6 bytes><...>) are sent to TCP/IP. The TCP/IP address is coded in BCD.
- Other NSAP or SNPA are sent to the OSI transport layer.

Internal Loopback

The session layer does a loopback when the calling and called NSAPs are equal. It is mandatory to have addressing profile using NSAPs.

CLNP also performs a loopback when the called NSAP is local.

Incoming Calls

Sharing of Communications Adapters

Physical devices are shared by several communications frameworks (OSI, TCP/IP, SNA, and so on). The next paragraph explains how the OSI Stack receives incoming data through LAN and X25 drivers.

You can find more information in *HiSpeed WAN Comm. Installation & Services Guide*, see Bibliography, on page B-1.

HiSpeed WAN Comm. Installation & Services Guide (86 A1 81WG)

LAN Networks

MAC frames are dispatched according to their format and to the destination LSAP and when the PDU does not contain LSAP, the PDU format is used (Ethernet / 802.3).

The OSI Stack receives the following LSAPs:

0xFE : ISO LSAP

configured inactive LSAPs (default is 0x20)

NetBIOS LSAP

The following multicast addresses are also predefined:

	ES–IS	**NetBIOS**
Ethernet	0x09 00 2b 00 05 0x09 00 2b 00 04	09 00 6H 00 01 00
Token Ring	C0 00 00 00 40 00 C0 00 00 00 80 00	CO 00 00 20 00 00
FDDI	90 00 14 00 00 20 90 00 14 00 00 A0	Not supported

X25 Networks

PVC are shared between communication stacks according to the static configuration. A PVC cannot be used by more than one stack. Each communication stack configuration (OSI, TCP/IP, SNA, ...) contains a list of PVCs (identified by a channel number) used on each line.

SVC dispatching at communications adapter level can be based on:

- received call user data
- called addresses and sub–addresses.

The OSI Stack requires X.25 device drivers to receive any incoming call regardless of call user data and called address. The actual dispatching depends on the X.25 device driver configuration.

Refer to individual X.25 communications adapter documentation to understand how dispatching can be configured.

For example, with HiSpeed Communications Adapters, the first dispatching criteria is the call user data. For the OSI Stack, a dispatching and called address can also be configured.

Dispatching Below Transport Layer

Figure 12 shows dispatching above the data link layer:

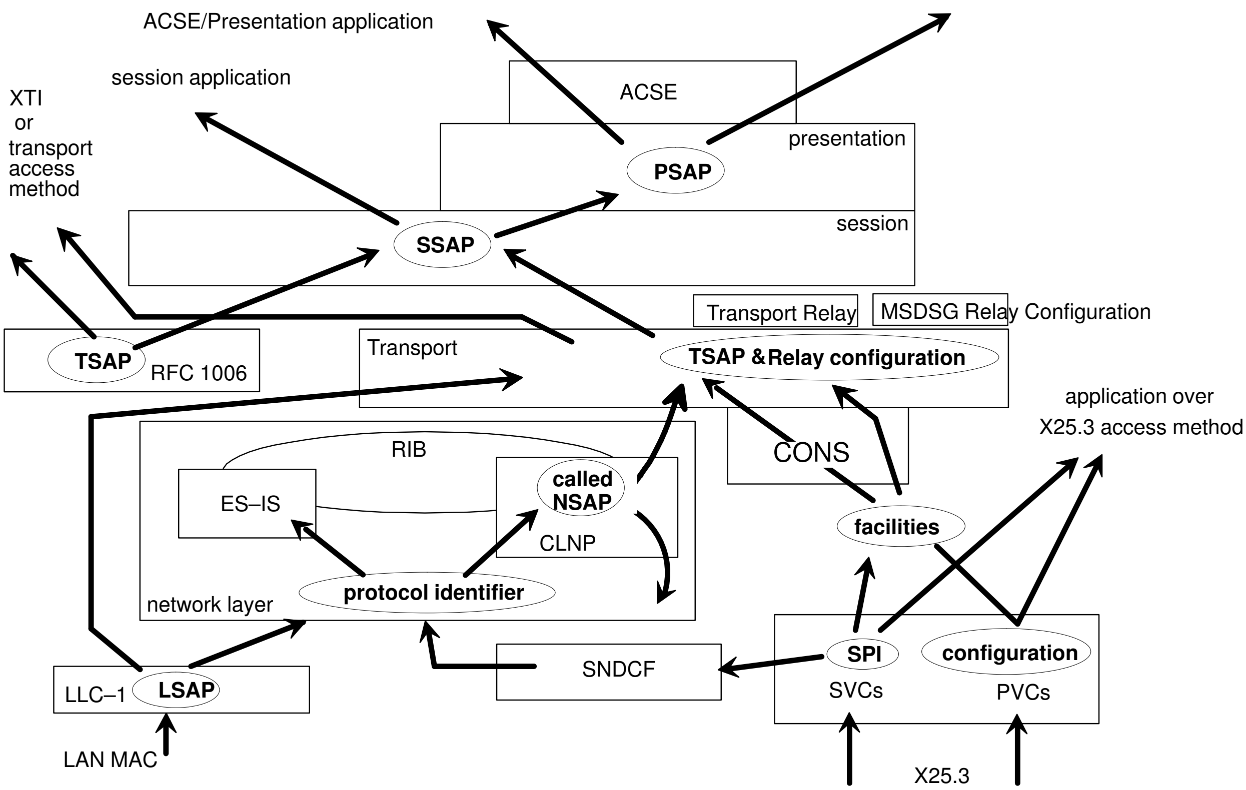


Figure 12. Dispatching Above Data Link Layer.

LAN Side

There are up to three dispatching levels:

- LSAP and protocol identifier are used by the LLC provider to distinguish between null CLNP and full CLNP profiles.
- Protocol identifier is used by the network layer to switch between ES–IS and CLNP.
- Called NSAPs are used to send PDUs to transport (for local NSAPs or remote NSAPs to be routed using MSDSG, a NetBIOS NSAP) or to activate the relaying function of CLNP.

The network protocol identifiers are defined by the CLNP and ES–IS protocols. They cannot be changed. The values are defined in ISO 9577:

- 0x00 null CLNP subset (usually called null IP)
- 0x81 CLNP
- 0x82 ES–IS

Behavior is not defined when several stacks or applications require to receive the same frame format and LSAP as the OSI Stack.

X.25 Side

The PVC configuration requires that a name (used by applications to identify a PVC) and the PVC owner be defined. The owner is either COTP or, the X25.3 access method for applications like TPAD–HPAD which are on top of this interface.

Data received on PVCs is dispatched within the OSI Stack according to the PVC configuration.

Incoming call packets (for SVCs), received by the OSI Stack, are dispatched according to the SPI value. The SPI configuration is used only to dispatch incoming SVCs. This configuration is global for the OSI Stack.

The SPI configuration allows a recipient for unknown SPIs and for incoming CALL packets which do not contain any user data (no SPI) to be defined, see default definition, SPI Table, on page 9-101.

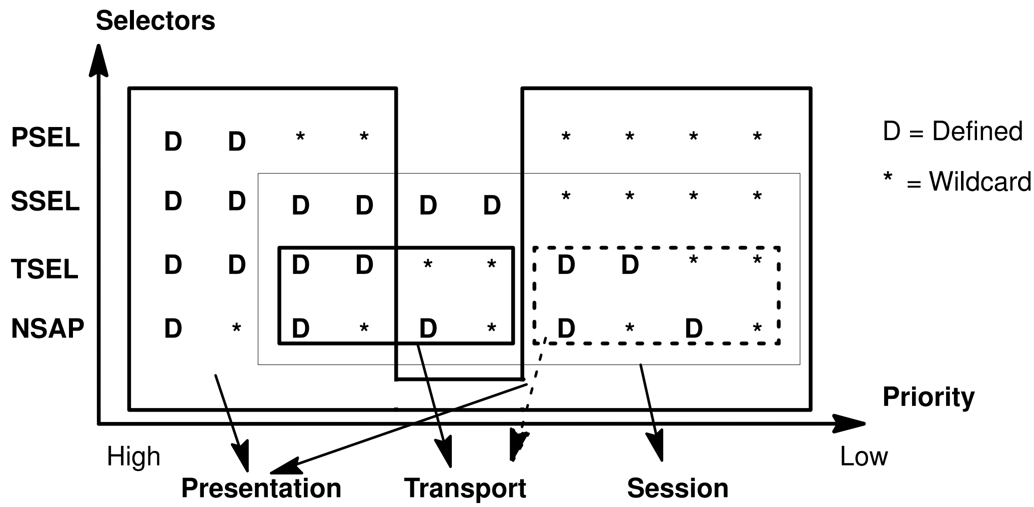
When CLNS over X.25 is used, the processing in CLNP is the same as for PDUs received on LAN subnetworks.

Addressing facilities allow the distinction to be made between profiles using a CONS service and those directly using X25.3 services. When address extension facility fields are found, a CONS profile is used.

Common Rules for Transport & Upper Layers

The transport, session and presentation providers implement wildcard addressing based on the called address. The priority rule used to solve conflict is consistent between each provider (i.e. the elements which are part of several rules have the same respective priority order).

The rules for determination of priority are shown in Figure 13.



Note: The higher selectors have the highest priority.
Presentation supports a subset of possible combinations.

Figure 13. Rules for Priority Determination.

Wildcard selectors are defined in Wildcard addressing Concepts, on page 3-3 and NSAPs are defined in Addressing Parameters, on page 3-6.

The wildcard TSEL value is one of the possible TSEL values. It is therefore not possible for applications to receive only connections with wildcards called TSELs. A bind with this TSEL will deliver a connection regardless of the TSEL value. This problem does not exist with the wildcard NSAP because it has a null length, and a real NSAP has at least one byte.

It is not possible to have several simultaneous binds on the same TSAP. Incoming connections are buffered on each TSAP (the limit is defined by the application).

To receive connections with profiles which do not transmit NSAPs (i.e. inactive CLNP of transport over X.25), is is mandatory to use the wildcard NSAP in the bind address.

The wildcard SSEL (4 bytes null) is one of the possible SSEL values. It is not possible for session users to receive only incoming connections with a 4 bytes null SSEL. This remark also applies for wildcard PSEL.

Special Support for TPAD/HPAD

The TPAD–HPAD product uses a private transport interface which requires a static configuration of TSEL for incoming connections using the **cftran tool**, see cftran Command, on page 11-2.

The called NSAP is not taken into account; it is similar to a bind with wildcard NSAP.

A default TSEL is reserved for TPAD–HPAD: 0x40,0x01,0x50,0x41,0x44,0x31.

When configuring TSELs for PAD, the administrator must check that they are not already in use; it is not possible to have the same TSELs for TPAD–HPAD and other applications.

Support of MSDSG Relay

The relay mechanism is shown in Figure 14.

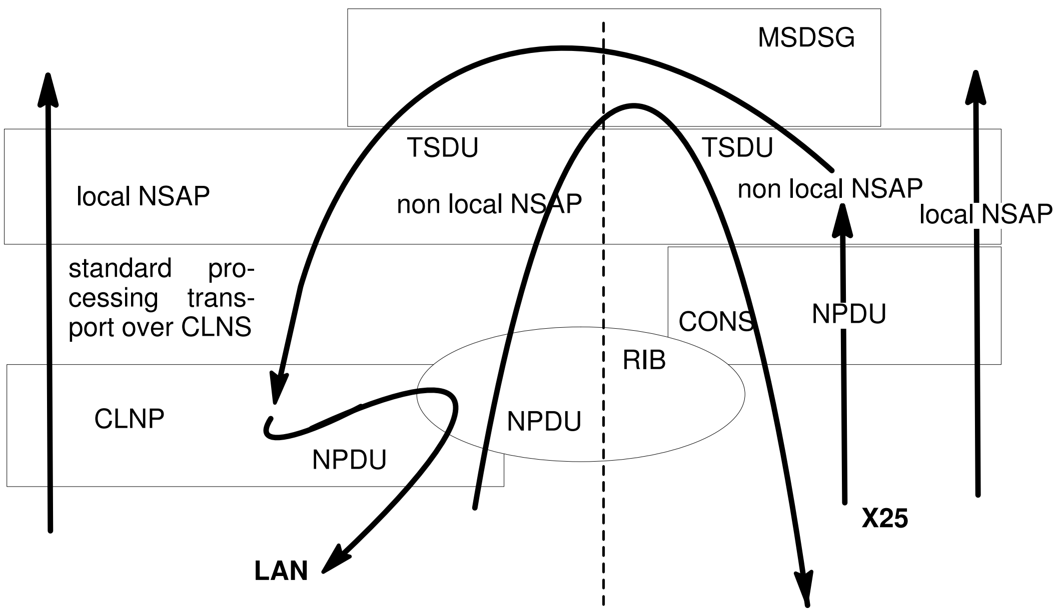


Figure 14. Relaying of MSDSG Incoming Calls.

CLNP tries to route the NPDU which called the NSAP (neither a local NSAP or the NetBIOS NSAP). If the called NSAP is found in the static RIB and the NSAP type of the selected RIB entry is MSDSG, the NPDU is sent to the transport layer and re–assembled. Received TSDU which are not destined to a local NSAP are delivered to the MSDSG relay. The relay established an outgoing connection using the CLCO profile, see page 3-14.Therefore the fixed part of the static RIB must contain all the NSAPs routed by MSDSG.

NPDU received on X25.3 are delivered to the transport layer and re–assembled. Received TSDU which are not destined to a local NSAP are delivered to the relay which opens a transport connection. When routing from X.25 to LAN, the called NSAPs are not necessarily in the static RIB because CLNP may use the dynamic RIB.

The MSDSG can operate regardless of the system type (ES, IS, or both).

Support of Transport Relay

The relay mechanism is shown in Figure 15.

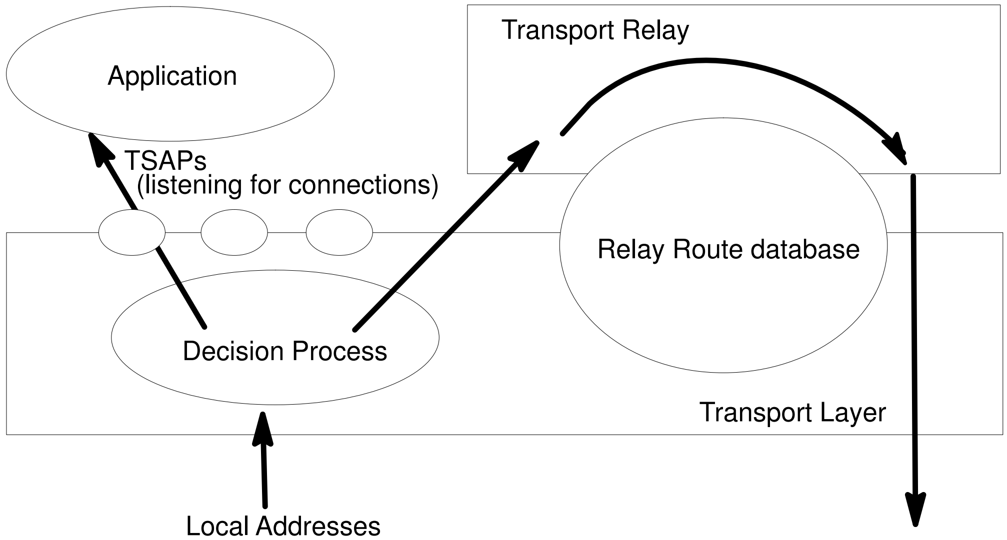


Figure 15. Relaying of MSDSG Incoming Calls.

The incoming transport connection requests are dispatched between applications waiting for connections and transport relay using this algorithm

if the called TSEL is one of a waiting application

then deliver the indication to the application

else

if a route with a non–wilcard incoming called TSEL matches the incoming call

then deliver the indication to the relay

else

if an application is waiting on the wildcard TSEL

then deliver an indication to this application

else

if a route matches the incoming call

then deliver the indication to the relay

else the called address is unknown (refuse connection).

The object of this algorithm is to give higher priority for local applications.

Special Rule for Session

The Session incoming calls are shown in Figure 16.

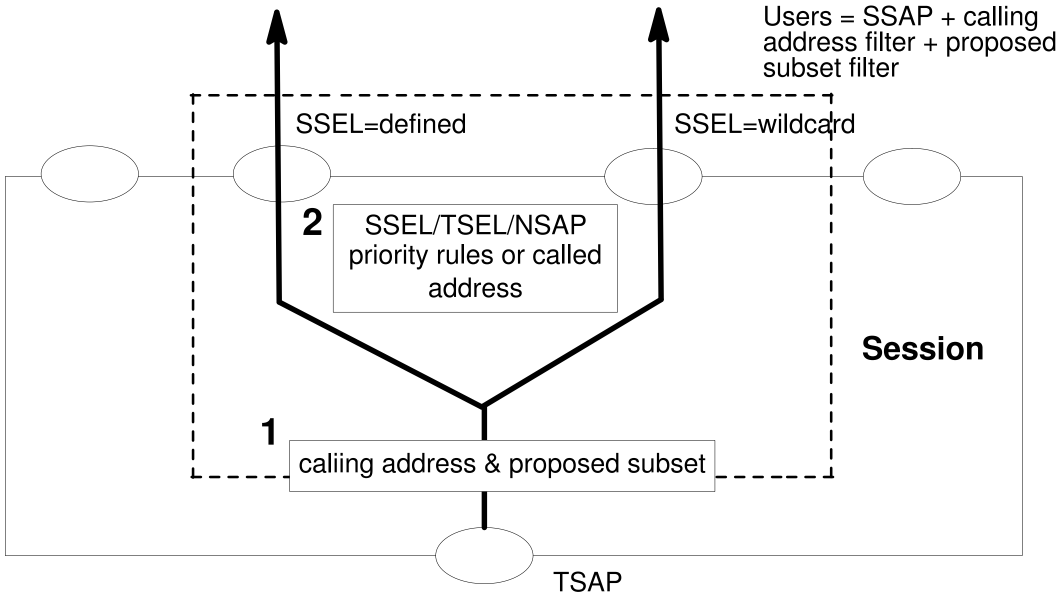


Figure 16. Session Incoming Calls.

Session users waiting for incoming connections may specify 3 types of information to filter incoming calls:

- calling address (SSEL, TSEL, NSAP or SNPA)
- proposed subset (BCS, BS, BAS, undefined)
- called address (SSEL, TSEL, NSAP). SNPA is not supported for called address.

The switch of incoming session connections is performed in two steps:

1. For each incoming session connection, only the listening users, matching the calling address and subset, are taken into account by comparing each address element which has a non null length. Null length elements are ignored. The subset proposed in the incoming session PDU is ignored if the user filters with the predefined value 'undefined subset'.
2. The common priority rule then applies on the called address to select one of the listening users not discarded by the matching of calling address and proposed subset.

Notes:

1. XAP applications do not use calling address and subset filtering.
2. Do not use this mechanism through ROSE API.

It is possible to have several session users bound simultaneously with the same requirements (SSAP, calling address, subset). The incoming connections are delivered to these users in the bind order; first connection to the first bound user, second to the second bound user, etc.. There is no queuing mechanism as for the transport provider.

Incoming session connections are stored by the session provider during a configurable delay, waiting for possible delivery to a new or free session user.

Chapter 4. Configuration & Limits

Tabulates configurable parameters with limits. Indicates whether they are configurable using the Configurator, or by Applications.

Configuration & Limits Overview

You can find more information in:

- For the following components, there are no specific limits:
  - security,
  - profile,
  - transport.

Conventions

The limits and configurable parameters are structured by components. For each component, several classes of parameters are defined:

- protocol parameters,
- addressing parameters,
- management parameters (parameters used for stack management and not already covered by the two first classes),
- implementation specific parameters.

All the classes are not pertinent for each component.

The following information is associated with each parameter:

- limit,
- configurability,
- comment.

The limit is defined by a maximum value, a list of values (separated by '/' or a range. When significant, the default value is described after a comma. N/A means not applicable.

When a parameter is configurable, the way to configure it is provided. Current solutions are:

- SMIT (configurable through stack configurator using SMIT interface),
- CLI (Command Line Interface, configurable through stack configurator but not integrated in SMIT),
- For other cases, refer to Technical Support.

ACSE

Protocol			
maximum number of connections	N/A		ACSE does not limit the number of associations, but COPP does.
use of X.410 mode	boolean	API	per association
FU selection	(1)	API	Only kernel FU is available through APIs
presentation context (PCI)	unlimited	API	PCI are defined per connection, including default context and ACSE context. There is no limit per connection.
transfer syntax for negotiation	unlimited	API	per presentation context
protocol version	1		
Addressing			
calling/called AE Title	any value supported	API	per association
AE title length	limited		the limit is due to the maximum size of the stream control block, which contains the AE Title and other informations
number of local AE Title	2048		1 per connection

Implementation specifics			
user data	limited		limit due to session provider

(1): available ACSE Functional Units (FU) are:

- kernel,
- authentication.

Protocol			
maximum number of connections	1..2048, 64	SMIT Other: ask Technical Support	
presentation context	unlimited	API	PCI are defined per connection, including default context. There is no limit per connection.
transfer syntax for negotiation	unlimited	API	per presentation context
default context name		API	1 per connection
presentation FU selected	kernel	API	only kernel FU is supported
use of X.410 mode	boolean	API	
protocol version	1		
Addressing			
calling/called PSEL	any value	API	per connection
PSEL length	0..16 bytes		
maximum number of local PSEL	2048		1 per stream (or bind)
maximum number of local PSAP	2048		1 per stream (or bind)
Implementation specifics			
user data	limited		limit due to session provider

Protocol			
maximum number of connections	1..2048, 64	SMIT Other: ask Technical Support	
COSP connection timer	160..20000, 160 s	SMIT Other: ask Technical Support	QOS parameter.
COSP disconnection timer	40..20000	SMIT Other: ask Technical Support	QOS parameter.
initial token assignment	for each token: initiator side, responder side, responder choice	API	per connection
FU selected or subset selected	(1)	API	per connection. Some FU are not supported through session API. Subset can be specified only through the session API.
use of segmentation	boolean	API	per connection. This parameter defines if a maximum TSDU size must be negotiated.
use of extended concatenation	boolean	API	
session version used	1/2/1+2	API	per connection
initial synchronization point serial number	0..999999	API	per connection
use of extended concatenation	boolean	API	per connection
TSDU size from initiator to responder	0..65535	Derived from API parameters	derived from proposed SSDU size
TSDU size from responder to initiator	0..65535	Derived from API parameters	derived from proposed SSDU size
calling user data	(2)	API	session API only
reuse transport connection	boolean	API	per session connection. session API only.

Addressing			
calling/called SSEL	any value	API	1 couple per connection
SSEL length	0..16 bytes		
maximum number of local SSEL	2048		
connection identifier	any value	API	per connection
connection identifier length	64+64+64 +4 bytes		
maximum number of local SSAP	2048		
Implementation specifics			
number of bind requests	4096		It is possible to issue many binds per stream, but only the last one is taken into account. 4096 simultaneous binds are supported, but the number of local addresses is limited to 2048. There is no relationship with the number of connections configured.
incoming connection retention time	1..20000, 5 s	SMIT	a specific mechanism allows incoming connections to be buffered, even if nobody is waiting for these connections.
number of stored connection requests	2048		
user data	0..512 bytes for V1 64 Kb–1 for V2 (2)	API	The user may limit independently the size of sent and received data (SSDU size from initiator to responder and SSDU size from responder to initiator). Other limits may be defined by protocol or PICS. See below.

(1): Supported (Functional Units) FU are listed in Connection Oriented Session Service Provider on page 2-5.

(2): The table overleaf provides more information about session user data size.

session service	session V1	session V2
connection establishment and release	512 B	10 KB (*)
data, typed data	64 KB – 1 (*)	64 KB – 1 (*)
capability data	512 B	64 KB – 1 (*)
expedited data	14 B	14 B
U–abort	9 B	10 KB (*)
P–Exception, P–Abort	0	0
Give token, Give control, Activity interrupt, Activity discard	0	10 KB (*)
other services	0, or 512 B	0 or 10 KB (*)

(*) The standard proposes an unlimited size, with restrictions documented in the PICS.

Session contexts and contexts linked to transport connections are dynamically allocated.

Protocol			
maximum number of connections	1.. 2048, 64	SMIT	
number of references	2*2048		No relationship with the maximum number of connections configured.
multiplexing rate (transport connection on VC)	1..2048	SMIT	The configurator ensures that the multiplexing rate is less than or equal to the maximum number of connections.
acknowledgment timer (AL)	0..2^{32-1}	Other: ask Technical Support	0 means no delay before sending acknowledgement.
retransmission timer (T1) on LAN	500..30 000 ms 3 000 ms	SMIT	
retransmission timer (T1) on WAN	500..60 000 ms 15 000 ms	SMIT	
number of retransmissions of a COTP LAN PDU (N)	1..15,10	SMIT	The configured value includes the initial transmission. It is therefore in the range 1..15, with 10 as default value.
number of retransmissions of a COTP WAN PDU (N)	1..15, 10	SMIT	The configured value includes the initial transmission. It is therefore in the range 1..15, with 10 as default value.
window timer (W)	N*T1 or 11 000.. 660 000 ms	Other: ask Technical Support	Computed value is not necessarily in the range of the configurable values.
add a different responding address	yes/no, yes	SMIT	Accept or not an incoming connection response which arrived on a different address than the connect address used.
time bound on reference and sequence number (L)	(N+1)* max (T1 LAN T1 WAN)		used only to freeze references. Freeze of sequence is not implemented. 25% of uncertainty for the real delay.
inactivity timer (I) on LAN	(N+1)* T1 LAN or 11 000.. 660 000 ms	Other: ask Technical Support	Computed value is not necessarily in the range of the configurable values.
inactivity timer (I) on WAN	(N+1)* T1 WAN or 11 000.. 660 000 ms	Other: ask Technical Support	Computed value is not necessarily in the range of the configurable values.
time to try reassignment / re–synchronization (TTR)	60s		

time to wait for reassignment / re–synchronization (TWR)	120s		
Supervisory timer 1 (TS1)	120s		
Supervisory timer 2 (TS2)	NA		not implemented
NSDU lifetime local–to–remote (MLR)	NA		should be integrated in the configuration of other timers
NSDU lifetime remote–to–local (MRL)	NA		should be integrated in the configuration of other timers
persistence time (R)	N*T1 on LAN / N*T1 on WAN		
transport class used	0/2/3/4, (4 with CLNS or CONS profiles, 2 with X25 profiles)	API	per connection
alternate transport class	0/2/3/4/none, none	API	per connection
use of expedited data	yes/no, (no for class 0 or yes for other classes)	API	per connection
non use of flow control	yes/no, (yes with class 0 or no with other classes)	API	per connection. Implementation allows to specify the opposite (use of flow control)
credit used	1..15, 15	API	per connection
TPDU size	128/256/ 512/1024/ 2048/4096/ 8192, (2048 with class 0 or 8192 with other classes)	API	per connection
preferred maximum TPDU size	NA		not implemented
non use of checksum	yes/no, yes	API	per connection. Implementation allows use of checksum to be specified.
communication profile		API	per connection. For some APIs, it is not possible to use the default value because the communication profile is used to determine the address format.
use of extended mode	yes/no, no	API	per connection
priority QOS	low/ medium/ high/ very high, low	API	
calling user data	any value	API	XTI API or TPI interface only

Addressing			
maximum number of local TSAPs	2048		1 per bind.
calling/called TSEL	any value	API	per connection
TSEL length	0..32 bytes		
maximum number of local TSEL	2048		1 per open
Implementation specifics			
interface segmentation (TIDU)	8100 Bytes		
number of open	2048		
number of binds	2048		1 per open.

(*) Internal contexts are dynamically allocated.

Some tables are allocated according to the previous limits.

MSDSG Relay

Addressing			
maximum number of routes	maximum number of static RIB entries (see RIB, on page 4-14.)		MSDSG routes are defined in the fixed part of the static RIB

CLNP & SNDCF

Protocol			
PDU lifetime	10	Other: ask Technical Support	Significant for dynamic routing. For static routing, lifetime is defined per route.
PDU reassembling	5s	Other: ask Technical Support	per subnet
number of VC	64	Other: ask Technical Support	SNDCF
inactive VC lifetime	300s	Other: ask Technical Support	SNDCF, per subnet
protocol version	1		
checksum generation	no	Other: ask Technical Support	full CLNP profile only.
select error report option	no	Other: ask Technical Support	
priority		API	4 priority levels are defined
protocol subset	full protocol/inactive	API	This parameter is derived from the communication profile selected.
subnetwork status	on/off, on	SMIT Other: ask Technical Support	per subnetwork (i.e. per communications adapter). Only for full CLNP profile.
Addressing			
calling/called NSAP	any value	API	per PDU
maximum number of local NSAP	32	Other: ask Technical Support	
NSAP length	1..20 bytes		
local NSAPs	any value	SMIT Other: ask Technical Support	NSAP Local or Group

ES–IS

Protocol			
system type	ES/IS/ES+IS	SMIT	
ES timer	1..20000, 180 s	SMIT	
IS timer	1..20000, 180 s	SMIT	
holding time	2*ES timer		the holding time is transmitted to remote systems through ESH PDUs.
checksum generation	yes		
configuration notification	yes		
options	ESCT		The End System Suggested timer is not transmitted, but the received value is used.
Addressing			
NET	any value	SMIT Other: ask Technical Support	first local NSAP
Management			
enable ES–IS		SMIT Other: ask Technical Support	

RIB

Implementation specifics			
number of static RIB entries	256..32737	Other: ask Technical Support	the limit may be increased but the whole static fixed RIB entry must be smaller than 600.
number of dynamic RIB entries for ES	256	Other: ask Technical Support	equal to the number of static RIB entries.
number of dynamic RIB entries for IS	8+1024*<number of communications adapters configured>		
waiting entries queue size	(1)		
number of variable static RIB entries size	(1)		
number of fixed static RIB entries size	1..static RIB –1, 200		
RIB entry size	104 bytes		
holding time refresh	60 s		the holding time of received entries is updated every 60s.

Note: (1): The size of the queues is defined by the rule:
fixed + variable + waiting = static.

LLC

Addressing			
calling/called LSAP		API	
LSAP to receive			per medium type.
maximum number of receiving LSAP for null CLNP profile	16	SMIT	
receiving LSAP for null CLNP profile	any value, excluding 0xFF, 0xFE	SMIT	Reserved LSAP as ISO LSAP or NSAP for Multicast Netbios preconfigured
LSAP for NETBIOS multicast	any value excluding 0xFF, 0xFE and the LSAP for null CLNP profile	SMIT	

LAN

This paragraph describes LAN mappers which are interface mappers between the OSI Stack and LAN device drivers.

The same limits and configurable parameters apply to each of the LAN mappers (i.e. Ethernet, Token Ring, FDDI).

Addressing			
calling/called SNPA		API	may be automatically generated when NSAP addressing is used.
maximum number of physical attachments	8 per type of medium		
physical attachments		SMIT Other: ask Technical Support	physical attachments are identified by a location code which identifies the slot on the which the communications adapter is located

X25

This paragraph describes WAN mappers which are interface mappers between the OSI Stack and WAN device drivers.

Protocol			
number of SPI	18		including no value and unknown value.
SPI	0..4 bytes	SMIT	use to dispatch VC in the stack.
default facilities		SMIT	some facilities may be forced by the X25 component. Facilities may also be added by protocol layers on top of X25.3 (CLNP/COTP).
facilities		API	per VC
call user data		API	per VC. Through all the APIs. CUD may also be defined by the Stack.
priority		API	per VC
Addressing			
number of HiSpeed WAN Comm.	0..6	SMIT	
maximum number of local DTE addresses	24		6 HiSpeed WAN Comm. Adapters (1 & 4 ports)
calling/called DTE address or PVC name		API	per VC
length of PVC name	1..8 ascii characters		
DTE address length	1..15 decimal digits		BCD coding

Implementation specific			
maximum memory allocated	1..1 000 000 Kb 2 Mb	SMIT	Value configurable in the range. 80 * number of COTP connections + 4096 * number of X.25 lines.

Chapter 5. Planning Your Installation

Planning Your Installation Overview

This section is dedicated to administrators who are not familiar with OSI providing guidelines to help you to install an OSI network and OSI Stacks.

You can find more information in:

- What to install?, on page 5-2.
- What is the licensing policy?, on page 5-4.
- How to interconnect stations and applications, on page 5-5.
- How to interconnect networks, on page 5-16.
- How to manage security, on page 5-23.

What to Install?

OSI Stack Bundles

Bundles are packages of telecommunications features grouped together with the OSI Stack. This approach provides easy installation and simplifies the use of telecommunications products.

The OSI Stack is integrated into a bundle which provides solutions for basic communications.

If you are planning to use these bundles, refer to the or *BASIC-COM-MP and GCOS-COM-MP Installation Guide*, see Bibliography on page B-1.

If you are not using bundles, you need to read the next paragraphs:

- Upper Layers, Lower Layers, Framework, on page 5-2.
- OSI Applications – OSI Layer Requirements, on page 5-3.
- OSI Lower Layers vs TCP/IP Layers, on page 5-3.
- Managing Inter–operability Constraints, on page 5-3.

CAUTION:
Some parameters, modified using the OSI Stack configurator, may be lost if the Bundles configurator is used subsequently.

Upper Layers, Lower Layers, Framework

Before starting your installation on a set of machines, you must determine what should be installed. The OSI Stack is made of three components (which are not always necessarily needed):

- OSI upper layers (session, presentation, ACSE),
- OSI lower layers (up to transport layer),
- OSI framework (administration and troubleshooting tools, common runtime for upper and lower layers).

A good approach is as follows:

- identify the machines which will host OSI applications,
- determine if these applications will use OSI lower layers (i.e. OSI network) or TCP/IP lower layers with NetShare (TCP/IP network), or both, remembering that there is no relaying between OSI and TCP/IP with the OSI Stack.
- determine if you have inter–operability constraints with other machines (PC, DPX/2, GCOS, or other),
- when an OSI network is preferred, identify which machines will be used for relaying between interconnected networks.
- OSI Lower Layers must be installed on these machines, even if there is no OSI application on the relaying system.

OSI Applications – OSI Layer Requirements

The list of OSI applications helps you to determine whether OSI upper layers should be installed. This information is available in each product manual, and are summarized below:

	OSI upper layers	**OSI lower layers**
X400, X500	no **(1)**	alternative **(2)**
TPAD–HPAD	no	yes
OTM, CPI–C SS	yes	yes
other BULL application	yes	alternative
XTI API	no	alternative
XAP, ROSE and session APIs	yes	alternative
X25.3 access method (API for X25.3)	no	yes
OSI relay	no	yes

Notes:

1. X400 and X500 have their own embedded upper layers.
2. Alternative means that TCP/IP and NetShare can be used instead of OSI lower layers.

The OSI framework must be installed at the same time as the upper or lower OSI layers.

OSI Lower Layers vs TCP/IP Layers

The choice between OSI lower layers or TCP/IP must take into account the following considerations:

- use of OSI lower layers is the ”natural solution” and use of TCP/IP is a new emerging way,
- OSI and TCP/IP stacks can coexist on the same station, but the administrator will have to manage two stacks,
- the use of TCP/IP or OSI is an end–to–end solution: it is not possible to establish a connection with OSI upper layers on top of TCP/IP on one side and OSI upper layers on top of OSI lower layers on the other side, so be careful about inter–operability problems,
- there is no OSI – TCP/IP relay in the BULL OSI Stack offer,
- it is easier to have a homogeneous solution on the whole network,
- some products need OSI lower layers.

This manual will not solve the question ”Is OSI better than TCP/IP”. If you choose to use TCP/IP, refer to the NetShare documentation, see Bibliography, on page B-1, for all that concerns the lower layers. It is always possible to migrate from one solution to another.

Managing Inter–operability Constraints

If you have to operate in a heterogeneous environment, you must identify a common subset of supported profiles. For example, use of TCP/IP by OSI applications may not be supported on other systems.

Refer to How to interconnect stations and applications, on page 5-5 to understand the notion of profiles.

What is the Licensing Policy?

The use of OSI upper or lower layers is dynamically controlled using iFOR/LS technology. You must therefore obtain and install licenses for each system which will host an OSI Stack. Each license contains a system identification and can therefore only be used on the system for which it has been ordered.

For details, refer to License Control – iFOR/LS, see page 7-9 and the Software Release Bulletin file delivered with the software.

How to Interconnect Stations & Applications

In order to interconnect applications you have to solve the following problems:

- be able to reach the station on which the application is located,
- on that remote station, be able to reach the right application,
- when you have reached the right application, define how to exchange information.

The following table provides a comparison with TCP/IP to help understand the scope of each potential problem.

OSI	TCP/IP
being able to reach the station on which the application is located	define IP addresses, select gateways and routing protocols
on that remote station, being able to reach the right application	define ports for each application
when you have reached the right application, define how to exchange information	choose TCP, UDP, ..., protocols

Station Identification

There are three ways to identify a remote station:

- logical name
- address of a communications adapter
- name of a station's access path.

Logical Name

The logical name is called the Network Service Access Point (NSAP). It is the equivalent in the OSI world of an IP address in the TCP/IP world. Each NSAP uniquely identifies a station. It is, however, possible to have several NSAPs on a single station. There is no relationship between local NSAPs and network attachment, unlike IP addresses which are tied to network attachments.

The format of an NSAP is defined by the international and national organizations which ensure a worldwide addressing plan. If you have a private network you are free to use your own NSAP format. NSAPs are represented as a string of hexadecimal digits with an even length of 2 to 40 digits.

The NSAP recommended syntax, which is derived from the US GOSIP specifications, is as follows:

fixed prefix	station identification	selector (1 byte) non–0 value

Even if you do not follow this format, it is better to have a consistent addressing plan, with structured NSAP syntax, instead of a random NSAP allocation.

Communications Adapter Address

The address of a communications adapter also uniquely identifies a station. The SNPA addresses are defined either by the communications adapter (for example, a LAN communications adapter has a pre–defined address), or by an organization (for example, when an attachment to the X.25 public network is rented, the organization in charge allocates an address).

Station Access Path

The station access path can be used only with permanent point–to–point connections. For example, when using PVC on X.25, the local description of the PVC uniquely identifies the station to which you are connected (there is no need to specify a remote address).

Although NSAP is the addressing format defined by ISO to identify stations, the other formats are also supported by the OSI Stack. Usually, when specifying an address for Stack or application configuration, the address format must be specified.

The table summarizes format and equivalent names.

Type	LAN	X.25
logical name	NSAP	
adapter address	MAC address	DTE address
path	not supported	PVC name

The term Subnetwork Point of Attachment (SNPA) is an OSI term to identify both MAC addresses, X.25 DTE addresses and PVCs.

You can find more information in Addressing, see page 3-1 and in ISO 8348, see Bibliography, on page B-1.

Profile Description

The Bull OSI Stack can be used in a large number of configurations and can communicate with other OSI–compliant machines using a variety of ”profiles”. A profile, in the OSI communications world, is a subset of the OSI protocols that specify one particular way that two machines can communicate. For example, an OSI profile might specify ”X.25, with transport class 2”, or it can specify how the Transport Layer can use the network services offered by a Token Ring network.

In this implementation, the profile is also a common way to specify the addressing format to use.

This section compares 7 profiles available in the OSI Stack. All these profiles, excepting that for TCP/IP, support both connection oriented and connectionless transport protocols.

For additional details, see OSI Telecommunications Profiles, on page A-1.

The profiles are defined for each connection by the application which establishes the connection. It is usually an application configuration parameter.

Check which profiles are supported by remote stations in the case of inter–operation with non–DPX/20 stations or non–Bull applications.

TCP/IP Profile

This profile uses the RFC 1006 implemented in the **NetShare** product in order to operate with the OSI Upper Layers on top of TCP/IP.

Description:	– stations are identified by NSAPs. – ConnectionLess network protocol is used on LAN & X25 network. – TCP/IP is used instead of OSI lower layers.
Advantage:	– you are able to perform relaying using TCP/IP relaying functions. – transparent use of LAN and X25 networks: from the application point of view, only the address value changes between LAN and X.25 networks.
Drawbacks:	– you may use connection less protocol over X25, which is not the best way to have good performances over X25. – NSAP format is predefined and integrate a TCP/IP address. – you are tied by TCP/IP addressing conventions configuration.

Full CLNP Profile

This profile uses OSI lower layers and LAN or X.25 communications adapters.

Description:	– stations are identified by NSAPs. – ConnectionLess protocol is used on LAN and X25.
Advantage:	– you can change the communication adapters without changing your addressing plan because there is no link between NSAP and SNPA values (except if you introduce a link through a NSAP naming convention). – transparent use of LAN and X25 networks: from the application point of view, only the address value changes between LAN and X.25 networks.
Drawbacks:	– you can use ConnectionLess protocol over X25, but this is not the best way to have good performances over X25. – PVCs are not supported. – requires configuration of routing information on X.25.

Equivalent terms: 3, Internet, CL, Full IP.

CONS Profile

This profile uses OSI lower layers and LAN or X.25 communications adapters.

Description:	– stations are identified by NSAPs. – Connection Oriented protocol is used on X25 and connectionless on LAN.
Advantage:	– same advantages as Full CLNP, resulting from using NSAP. – better performances over X25 thanks to the use of Connection Oriented protocol.
Drawbacks:	– X25–80 and PVCs not supported. – requires configuration of routing information on X.25.

Equivalent terms: 5, SPEE, CLCO.

Note: Full CLNP and CONS are equivalent on LAN.

Null CLNP Profile

This profile uses OSI lower layers and LAN communications adapters.

Description:	– stations are identified by SNPA. – ConnectionLess protocol is used on LAN.
Advantage:	– you do not have the overhead of the full CLNP protocol.
Drawbacks:	– supports only LAN networks. – you cannot define an addressing plan. because you are not free to choose SNPA addresses. – you must change application configuration as soon as you change a communications adapter.

Equivalent terms: 4, Ethernet, LAN, Null IP.

Configurable Null CLNP Profile

This profile uses OSI lower layers and LAN communications adapters.

Description:	– similar to Null CLNP, but uses configurable LSAPs. The default value is 0x20 to allow GCOS inter–operability on LANs.
Advantage:	see Null CLNP
Drawbacks:	see Null CLNP

Equivalent terms: 2, RLE, LANC.

X.25 Profile with SVCs

This profile uses OSI lower layers and X.25 communications adapters.

Description:	– stations are identified by SNPA. – Connection Oriented protocol is used on X25.
Advantage:	– use of SVCs
Drawbacks:	– supports only X25 networks. – you cannot define an addressing plan because you are not free to choose SNPA addresses.

Equivalent terms: 1, X.25, SVC.

X.25 Profile with PVCs

This profile uses OSI lower layers and X.25 communications adapters.

Note: There are no restrictions to the PVC format.

Description:	– remote stations are identified by a local logical name. – Connection Oriented protocol is used on X25.
Advantage:	– use of PVCs
Drawbacks:	– supports only X25 networks.

Equivalent terms: 6, PVC.

Stack Addressing Configuration

This paragraph describes which addressing information must be configured on stations which host OSI applications. To establish a relation between two applications located on different stations, you usually have to define parameters or make choices on both stations:

- the calling application must know how to reach the called application
- the called application must know how to respond to the calling application.

Figure 17 describes which information has to be configured, according to administrator choice or network considerations.

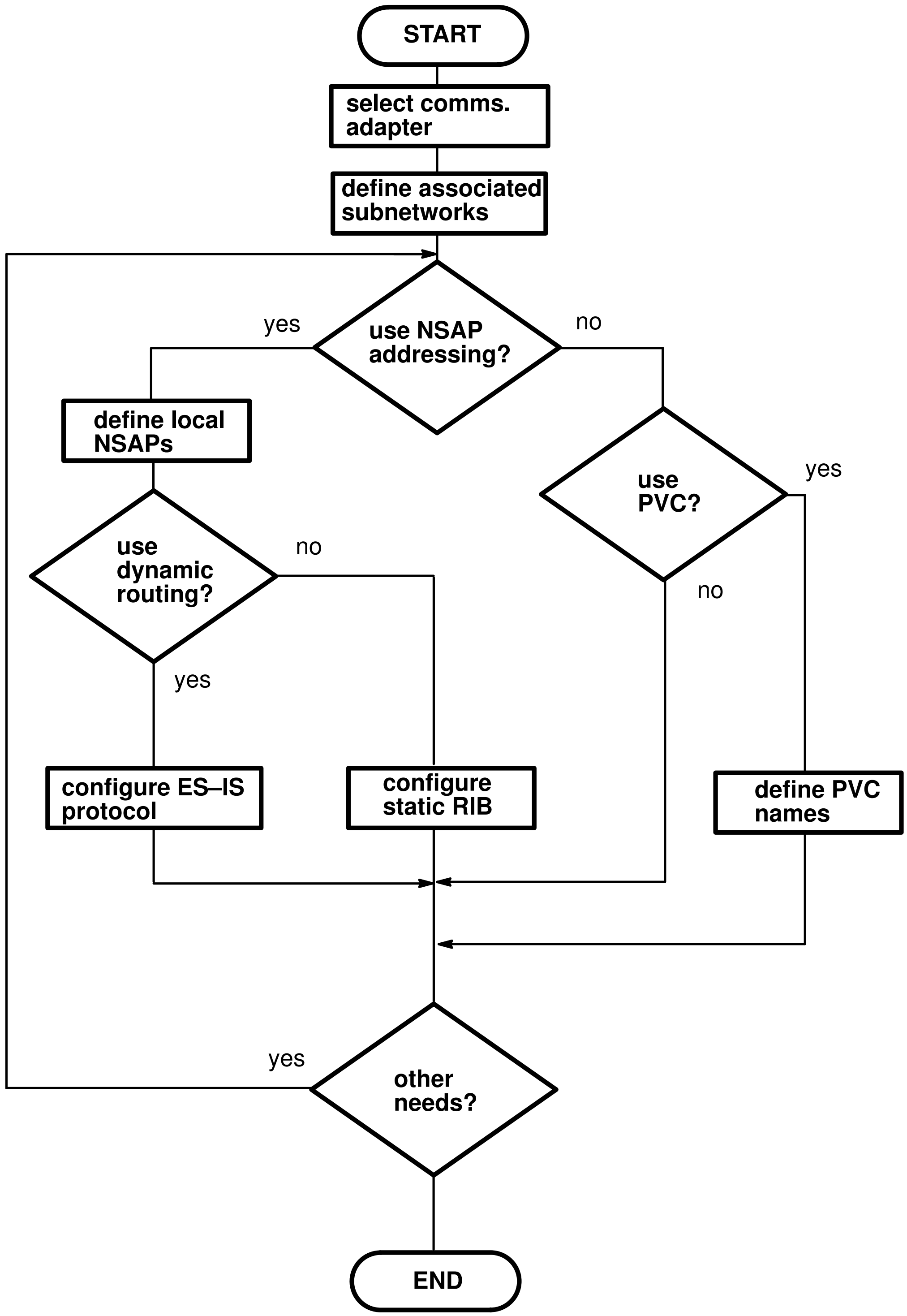


Figure 17. Choice of Stack Addressing Configuration

The communications adapters can be shared with other communications protocols (TCP/IP, SNA, IPX/SPX, ...).

The subnetwork is an object which allows a logical name (subdomain) to be associated with a network access. The use of a subdomain to identify a network access avoids defining it in the address, which is discovered automatically. The subdomain is a hexadecimal value which has only a local meaning and is never transmitted through the network. It is specific to the Bull OSI Stack configuration.

As soon as NSAP addressing is used, it is necessary to configure the OSI Stack to allow it to do a mapping of calling and called NSAPs on calling and called SNPAs, which are the only addresses known by the physical network.

Two options are available:

- static routing,
- dynamic routing.

The dynamic routing is based on the ES–IS protocol and is limited to LAN networks (because it requires multicast capabilities). It allows an automatic mapping of NSAPs to SNPAs with some limitations in case of LAN interconnections. To work properly, the ES–IS protocol must be active on both the calling, called and intermediate systems. See Use of LAN Subnetworks, on page 6-2. This information is stored in a dynamic Routing Information Base (dynamic RIB).

The static routing is available on LAN and X.25 networks and does not have any limitation. It is able to take into account addressing plans to reduce configuration overheads. The principle is very simple:

- for each remote NSAP (or group of remote NSAPs), define the subdomain to use (select the communications adapter to use).
- then tell the remote SNPA, which identifies the destination station where it is located on the same network, or a relay system in case of network interconnection.

This information is stored in a static RIB.

An example of static RIB configuration is shown in Figure 18.

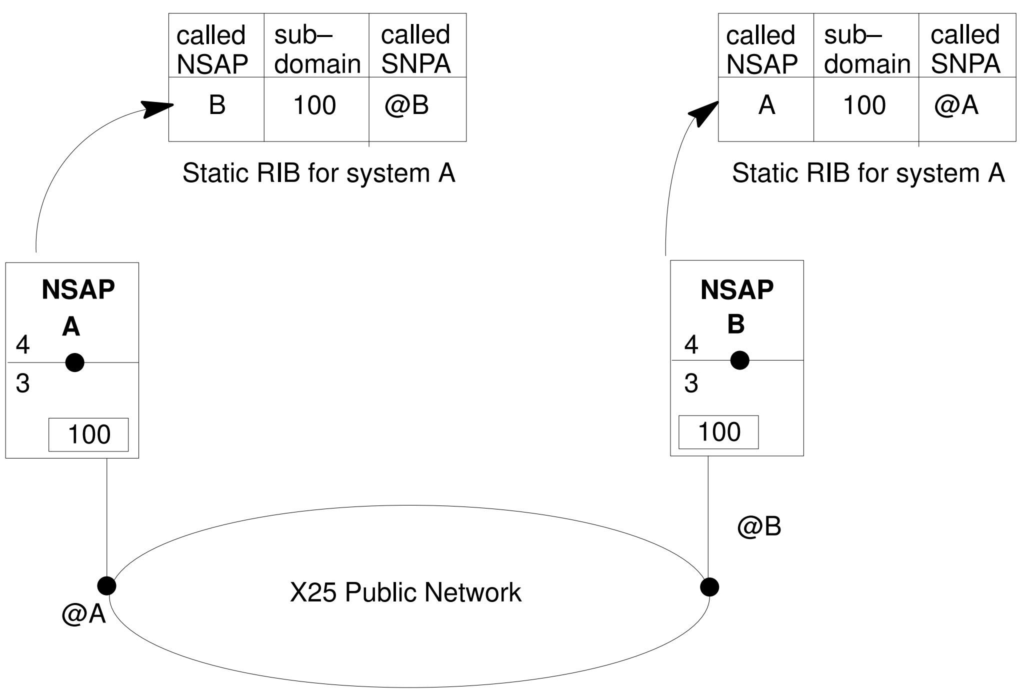


Figure 18. Stack Addressing Configuration

More information can be found in:

- Addressing, on page 3-1.
- OSI Stack Configurator, on page 9-1.
- Configuration Commands, on page 10-1.

Application Identification

Once you have selected the profile used and you have configured the necessary stack addressing parameters, you are able to reach any station you want. But now you have to reach the right application on that station.

With TCP/IP you have only a transport layer and the switch between applications is done on the port value. OSI has defined additional layers (session, presentation, application). The OSI equivalent for the port number is a selector and there are selectors for session and presentation layers. Above presentation layer, applications are identified by an Application Entity Title. Therefore, when an application wants to reach a remote application, it must provide all the local and remote selectors and AE Titles. Switching can be made at every layer, according to the selector used.

Do not panic with all these selectors: applications provide default values which simplify the configuration in most cases. If you have to configure this information yourself, you can apply the following rule: as soon as two applications use different selectors (or Application Entity Title) at the same layer, you are able to correctly switch the connections.

Be careful with applications like X.400 and X.500 which have their own embedded upper layers and are plugged on top of OSI transport layer: in these cases only the transport selector is meaningful.

If you need more information about OSI Stack addressing, refer to Addressing, on page 3-1.

A comparison between TCP/IP and OSI addressing is shown in Figure 19

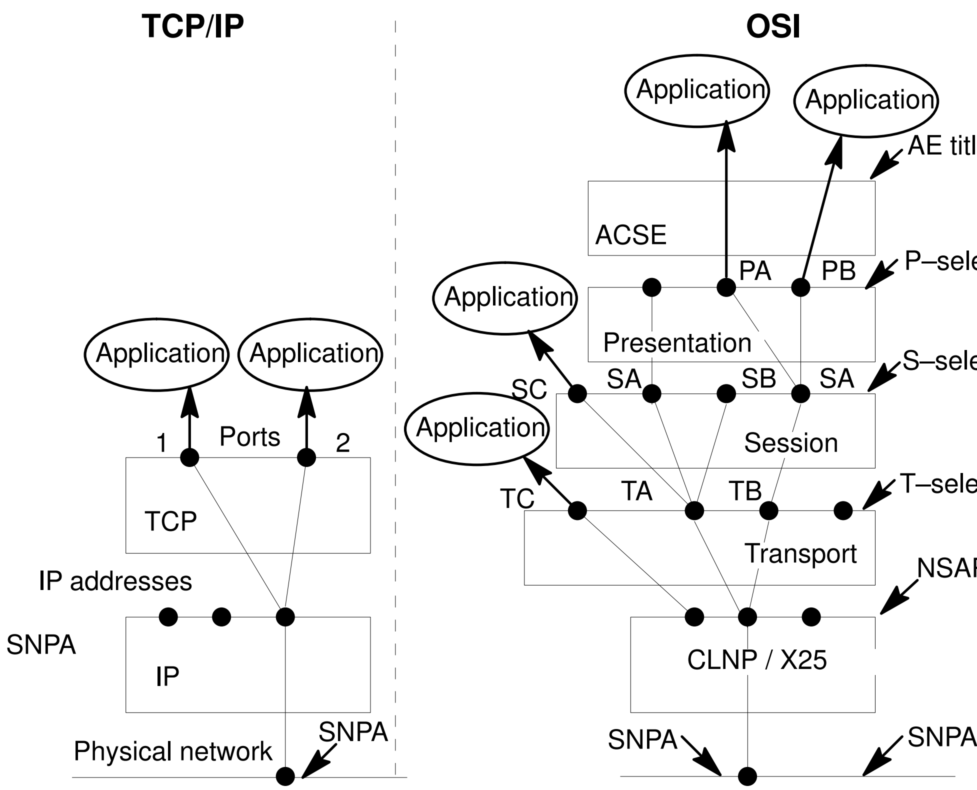


Figure 19. Comparison Between TCP/IP & OSI Addressing

Note: Switching can also be performed according to the NSAP, but it is not recommended.

Protocol Parameters Configuration

Once you are able to reach the right application, OSI offers many options on how to exchange information. Profiles define which protocols are used, but their parameters must also be configured:

- transport layer provides options to improve the quality of service according to the network used, and provides two data flow rates (normal and express)
- session layer offers many optional services to structure the information exchange and if you build your own application, you may have to select these options,
- presentation layer provides a means to be independent of machine representation (through abstract syntaxes and transfer syntaxes).

When operating over X.25, it may also be necessary to specify additional parameters, for example ”facilities” or ”call user data”. These optional parameters are reserved for advanced use, see Use of X.25 Subnetwork, on page 6-5.

The transport options are relevant for all applications. Session and presentation options usually depend on the application used.

There are two ways to configure the transport and X.25 parameters:

- using the Application configuration. Refer to the configuration procedures in the Application’s user manual.
- using the stack profile configuration for XTI or transport relay. In this case, profile descriptions are attached to a remote address (NSAP, SNPA) or to a PVC. The remote address is called **route.** Both route and profile descriptions are stored in a ”profile database”.

This manual will therefore focus on guidelines for the values of transport parameters in OSI layers. In the case where TCP/IP profile is to be used, refer to the *NetShare User’s Guide*, see Bibliography, on page B-1.

Transport Class

Description:	0: the simplest one for connection oriented networks. 1: not supported (recovery from a network disconnect or reset). 2: like class 0, but with multiplexing facilities. 3: like class 2, plus recovery from a network disconnect or reset. 4: like class 3, plus capability to detect and recover from errors (even if they are not notified); for connection oriented and connectionless networks.
Recommended value:	4 (mandatory) with CLNP protocol (and CL CLCO LAN LANC profiles). 2 with SVC or PVC connection oriented protocol over X25, or, 0 with SVC if multiplexing is not required on X25.

Note: Class 3 and 4 increase the memory use because the transport layers stores data to re–transmit them in case of failure.

Alternative Transport Class

Description:	if the transport class is refused by the remote transport user, propose another value.
Recommended value:	Not significant with CLNP protocol (and CL CLCO LAN LANC profiles). 0 for connection oriented protocol on X25.

Note: When the parameter is not significant, any value can be used.

Expedited Data Transfer

Description:	if this option is selected, the transport will provide two data flow rates. The express data may outrun normal data.
Recommended value:	depends on your application.

Expedited data is transmitted faster than normal data and has a separate flow control. Enabling or disabling this service does not impact performance.

Use of Flow Control

Description:	optional use of flow control mechanism. Use is forbidden for class 0, recommended for class 2, mandatory for classes 3 & 4.
Recommended value:	yes, except for class 0.

Initial Credit

Description:	number of TPDU stored waiting for flow control acknowledgment.
Recommended value:	maximum value (15) to improve performance.

When this value is decreased, memory use is decreased. The throughput is also increased as a result of better use of pipelining on the connection.

Use of Checksum

Description:	does the TPDU include a checksum for error detection with class 4
Recommended value:	No

The use of checksum reduces throughput because of the cost of checksum calculations.

TPDU Size

Description:	defines the maximum size of PDUs exchanged by two transport providers. This size corresponds to the transport header and the user data. Must be a power of 2 in the range 128 to 8192.
Recommended value:	the highest supported for the profile selected and the class used (i.e. 2 Kb if Class 0 is proposed, else 8 Kb.

High values increase the throughput, but also increase memory use.

Use of Extended Format

Description:	defines how TPDUs are numbered
Recommended value:	No Yes, in Class 4.

Priority

Description:	defines the priority of connections for data transmission and reception. Possible values are maximum, high, medium and low.
Recommended value:	Low

Increasing the priority should increase the throughput on the connection (compared to that on other connections).

Alternative Retransmission Algorithm

Description:	permits the use of a different algorithm for the retransmission of Data TPDU. THis can be useful for interoperability with some other implementations or when using interactive applications (like VTtelnet) exchanging small packets of data (1 byte = 1 TPDU). This is meaningful only in class 4. This parameter can only be activated using route and profile descriptions.
Recommended value:	No

Notes:

1. In Class 2, if expedited data is used, flow control is required.
2. If flow control is not used with Class 2 and there is multiplexing, the connection will be closed by the transport in case of overload.
3. Extended format is only supported in Class 4.

How to Interconnect Networks

This section explains how to interconnect physical networks and what special configuration may be required.

MAC Bridges

You can interconnect networks using MAC bridges. They are fully transparent from the OSI Stack point of view.

Network and Transport Relays

If MAC bridges do not solve your problem, you must use the OSI Stack network relay or transport relay facilities to relay connection oriented transport protocol.

Three relays are supported by the OSI Stack:

- Network relay (or CLNP relay)
- MSDSG relay
- Transport relay.

The network relay (or CLNP relay) is able to relay connection less protocol between any couple of LAN or X25 networks: it can be used for transport connections established with Full CLNP profile or for connectionless transport protocol. **The use of NSAP addressing is mandatory.**

The MSDSG relay provides the same services as the network relay, but it is also able to perform connectionless/connection oriented network relaying between LAN and X25. It is recommended for connectionless/connection oriented (CLNS/CONS) relaying or connection oriented (CONS/CONS) relaying. **The use of NSAP addressing is mandatory.**

The transport relay performs relaying between any LAN or WAN profile. **The use of NSAP addressing is not mandatory.** It also allows transport connection parameters to be modified. For example, to specify X.25 facilities or to modify the transport class when relaying from LAN to X.25. This relay is not transparent from an addressing point of view.

The next table summarizes which relay is recommended according to the profile selected.

Outgoing Profile	Full CLNP on LAN or X.25	CONS on X.25	Other
Incoming Profile			
Full CLNP	CLNP	MSDSG	transport
CONS	MSDSG	MSDSG	transport
Other	transport	transport	transport

All the relays can operate simultaneously on the same station on condition that there is no ambiguity on the manner in which to route.

Figure 20 describes how relaying is performed in the case of the CLNP relay.

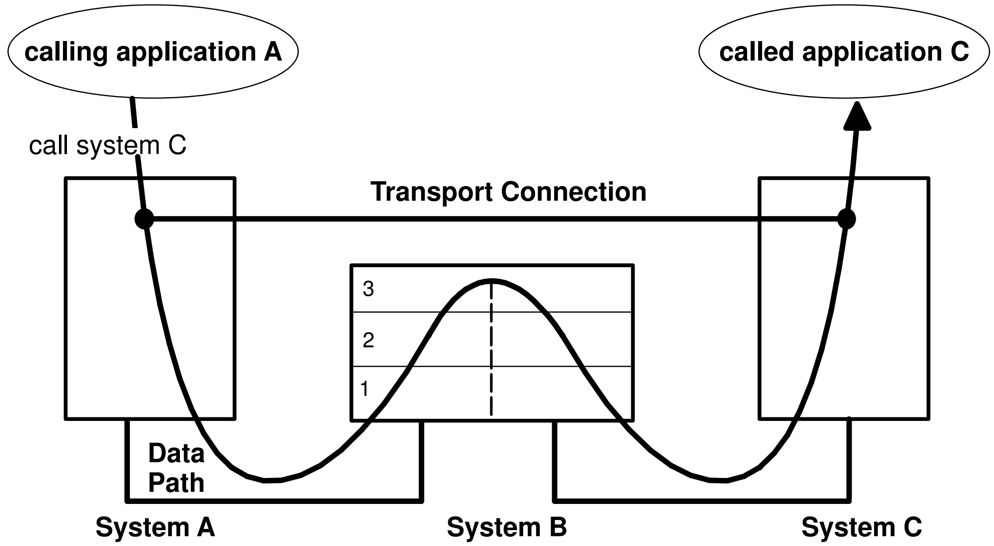


Figure 20. Example of CLNP Relay

An application on System A calls application C using the NSAP of System C as the called address. The relay, System B, is transparent for application A and application C. The routing criteria is the called network address.

Figure 21 describes how relaying is performed in the case of the MSDSG relay.

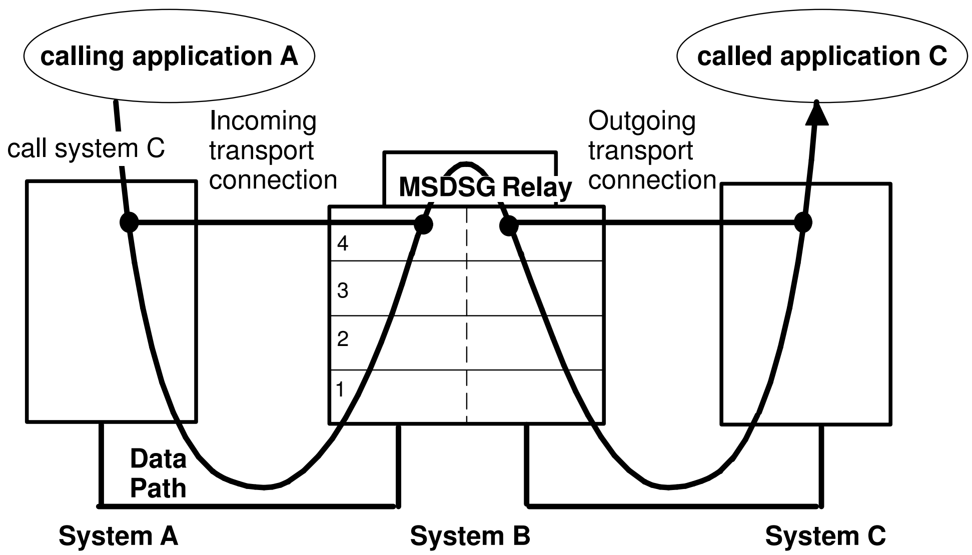


Figure 21. Example of MSDSG Relay

Application A calls application C using the NSAP of System C as the called address. The relay, System B, is transparent for application A and application C. The routing criteria is the called network address.

Figure 22 describes how relaying is performed in the case of the transport relay.

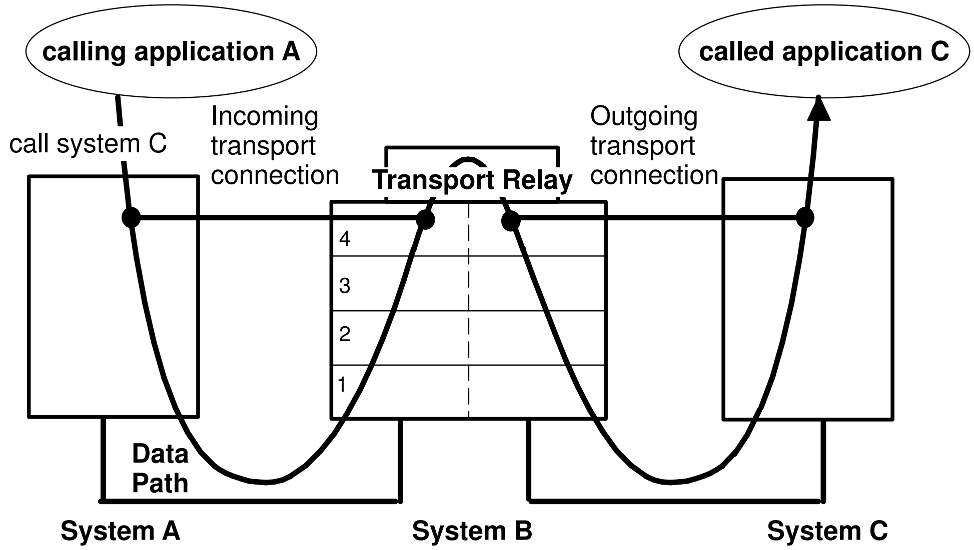


Figure 22. Example of Transport Relay

Application on System A calls application C using the address of System B as the called address. The relay, System B, is not transparent because application A calls System B. The routing criteria is usually the called transport selector.

Relay Configuration

The use of the three relays, CLNP, MSDSG and Transport, may require some changes to the configuration.

Figure 23 explains which actions have to be performed.

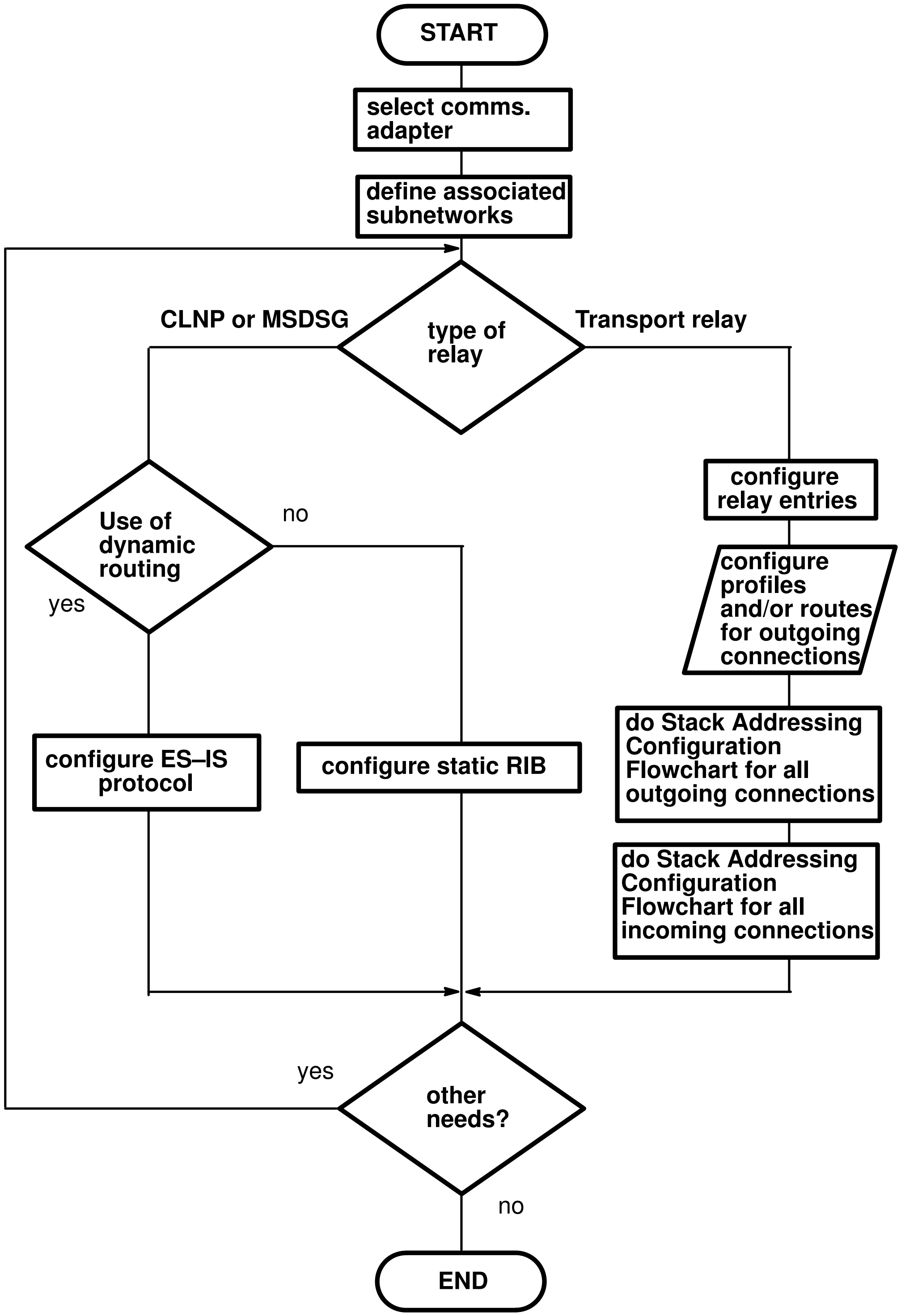


Figure 23. Relay Configuration Flowchart

See Stack Addressing Configuration flowchart, on page 5-10.

The use of full CLNP profiles on LAN may simplify the configuration of relay systems; they only have to be configured as Intermediate Systems (IS). See Use of LAN Subnetworks, on page 6-2 to understand the limitations of this solution.

Figure 24 shows a simple network using CLNP relay in conjunction with ES–IS protocol.

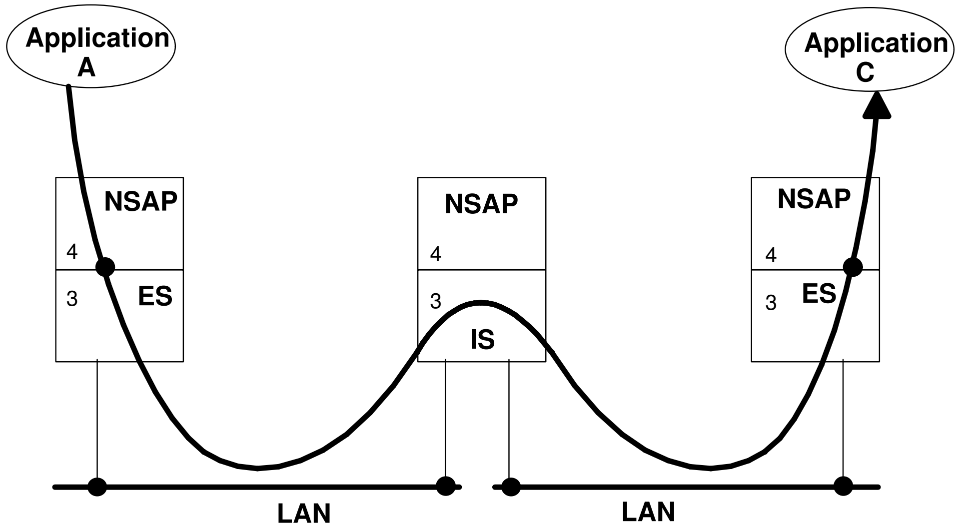


Figure 24. Use of Full CLNP Profiles for LAN Relays

The transport relay uses a specific database that must be configured. Each record contains addressing information about the incoming connection and data to establish the outgoing connection.

Figure 25 shows an example of network configuration using a transport relay. The calling application uses a ”null CLNP” profile on LAN and the called application requires an ”X25 with SVCs” profile. There is no need to configure static RIB, because NSAP addressing is not used.

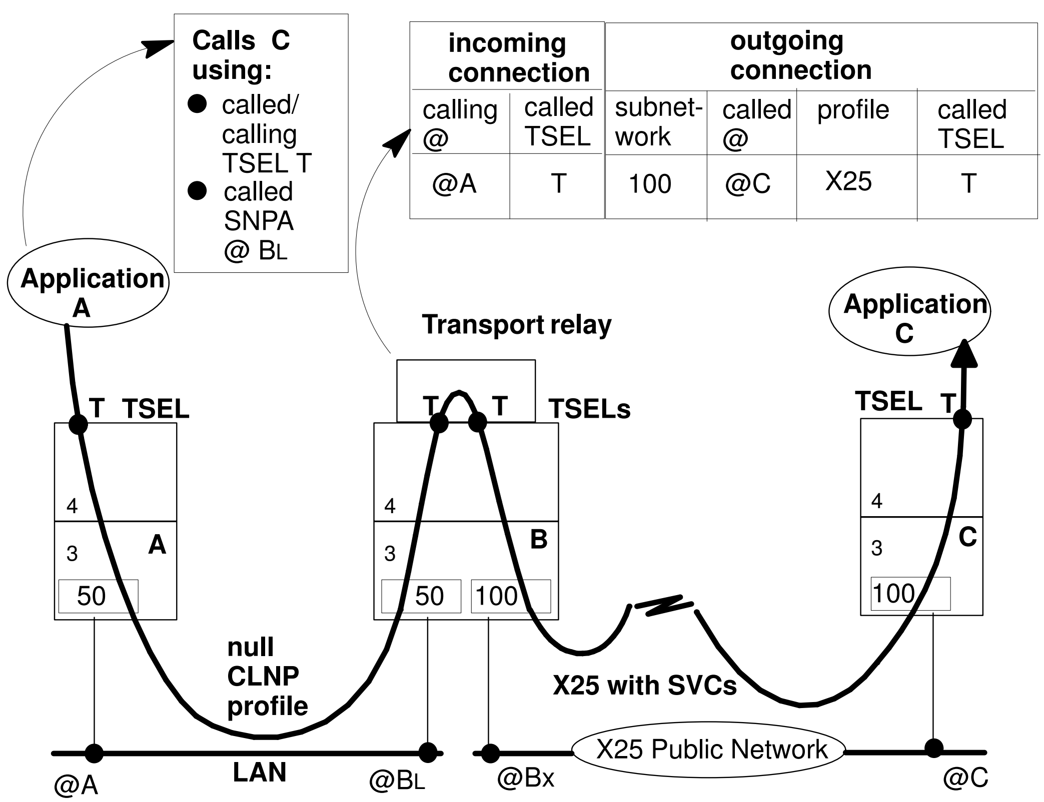


Figure 25. Example of Relay and Calling Station Configuration

This configuration allows Application A to connect to Application C. Only a subset of transport relay parameters are shown.

If the protocol between System B and System C are full CLNP on X.25, it would be necessary to also configure static RIB on systems B and C: in this case the transport relay is similar to an OSI application and requires the same addressing configuration.

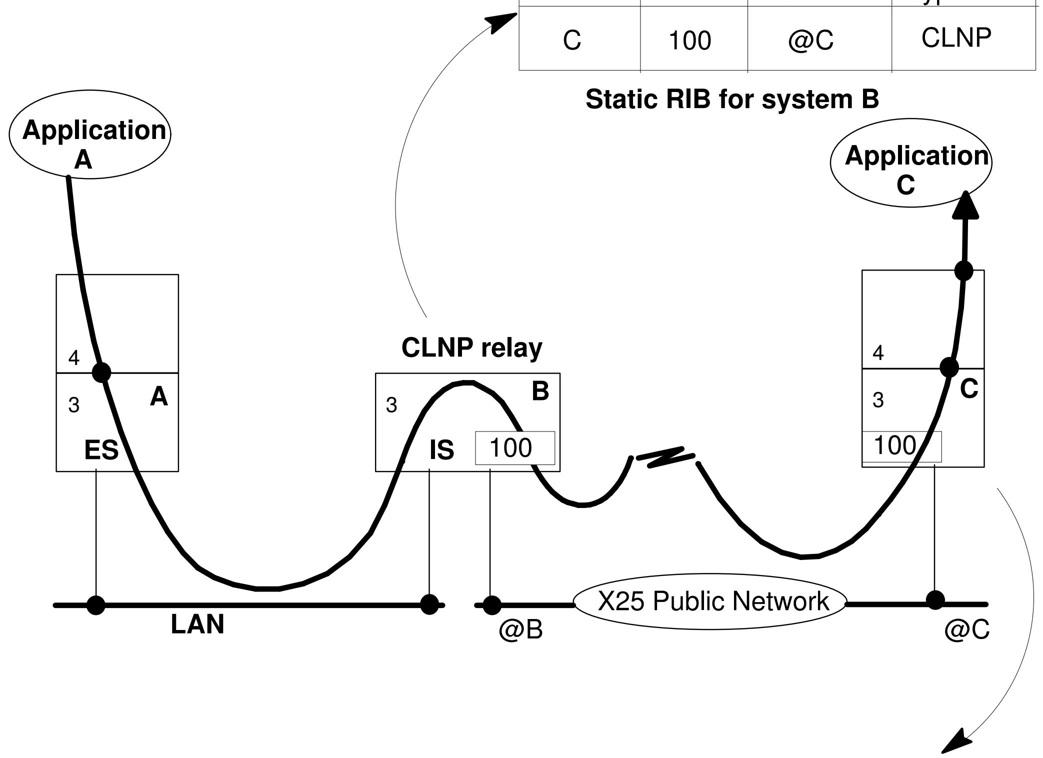
The transport relay uses information stored in the profile database (a default value) to select all the parameters mandatory for the establishment of the outgoing connection.

The CLNP or MSDSG relay may need to configure the static RIB to route incoming data. As explained in ”Stack Addressing Configuration” on page 5-10, the aim is to derive the local and remote addresses from the called NSAP. When relaying is used, the RIB has an additional function, to select the relay (CLNP or MSDSG). This is done using an extra parameter describing the relay used. The parameter is significant only when the incoming protocol is full CLNP.

Figure 26 shows an example of CLNP relay configuration. The relay station is configured as an IS to collect PDUs to be routed. It does not know the NSAP of system C because ES–IS protocol does not operate on X25.

called NSAP	sub domain	called address	relay type
C	100	@C	CLNP

Static RIB for system B



called NSAP	sub domain	called address	relay type
A	100	@B	Not Signifiant

Static RIB for system C

Figure 26. How to Configure CLNP Relay and Calling Station

In order to use the MSDSG relay and a CONS profile over X25, the relay type must be set to MSDSG.

Figure 27 shows another example, where the LAN/X.25 relay cannot be configured as an IS (because there is already an IS on the LAN) and where the MSDSG relay is used.

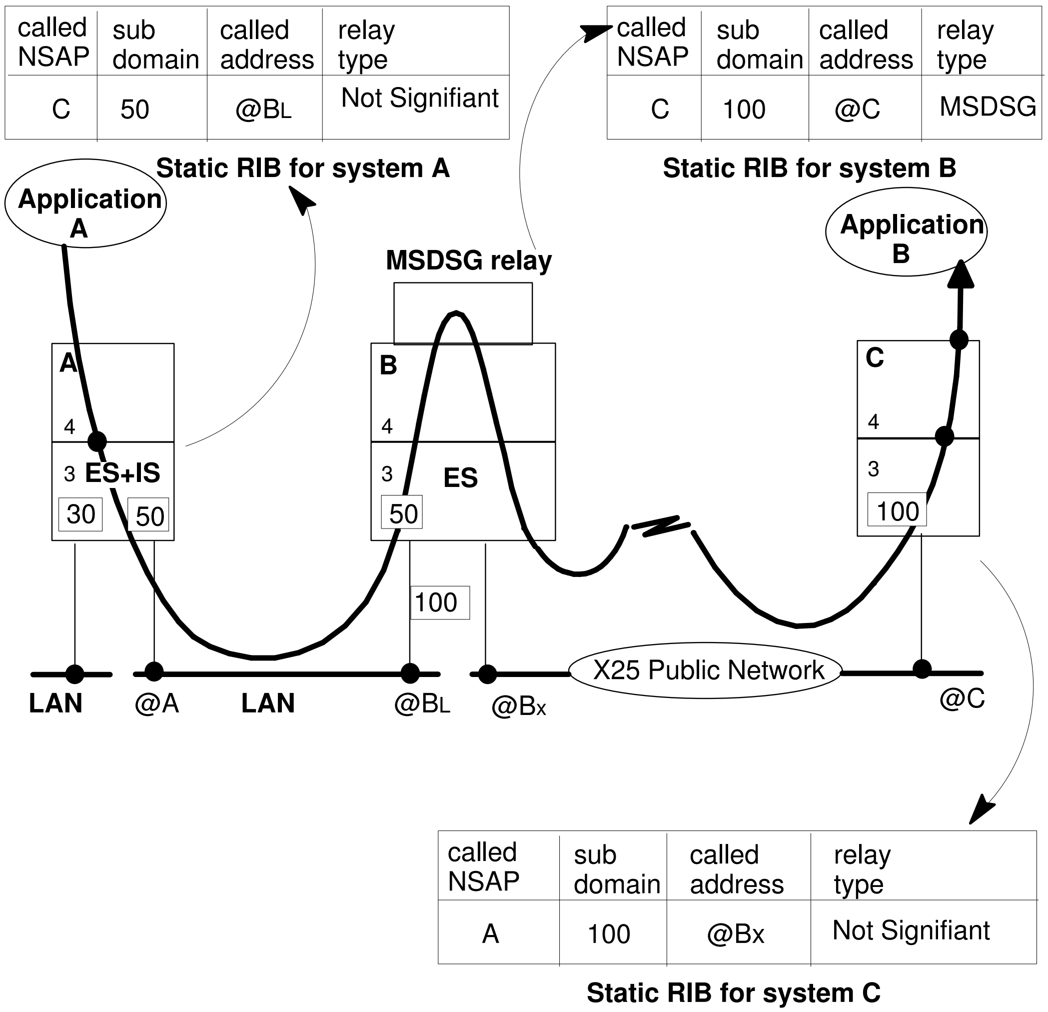


Figure 27. How to Configure MSDSG Relay and Calling Station

A static RIB entry is mandatory on system A to send PDUs for system C to the relay (which is an ES and not an IS, as on Figure 27). Thanks to the ES–IS protocol, there is no need to define a RIB entry on the relay to route from the route from system B to system A.

How to Manage Security

The OSI Stack offers a service to control the incoming data on system: the administrator defines a list of authorized calling and called addresses, and any data stream which is not in the lot of analyzed addresses is refused. This mechanism applies both for the data stream to the application located on the system and the data stream relayed to another system. There are two control points to cover both streams, as shown in Figure 28.

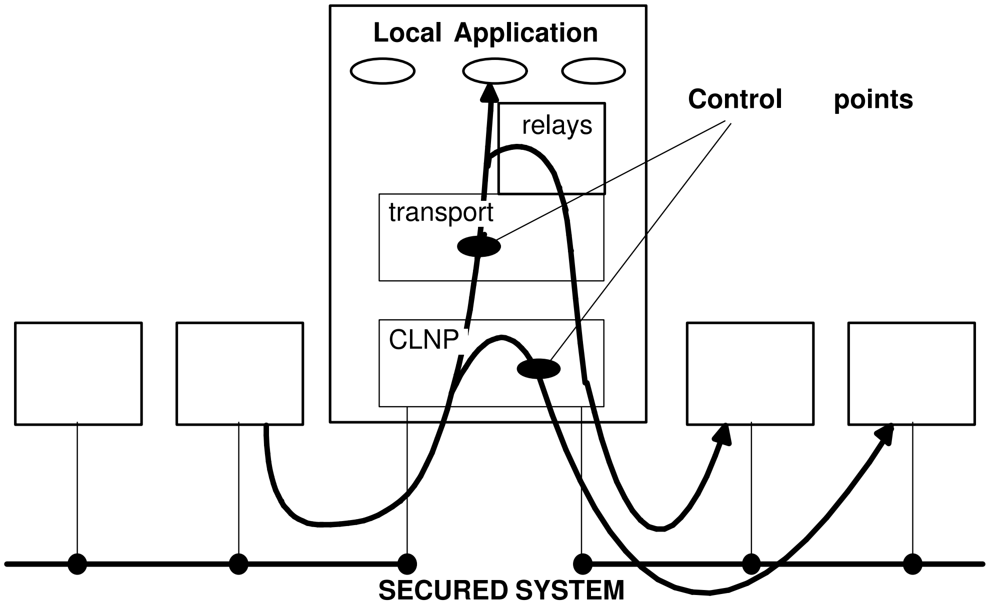
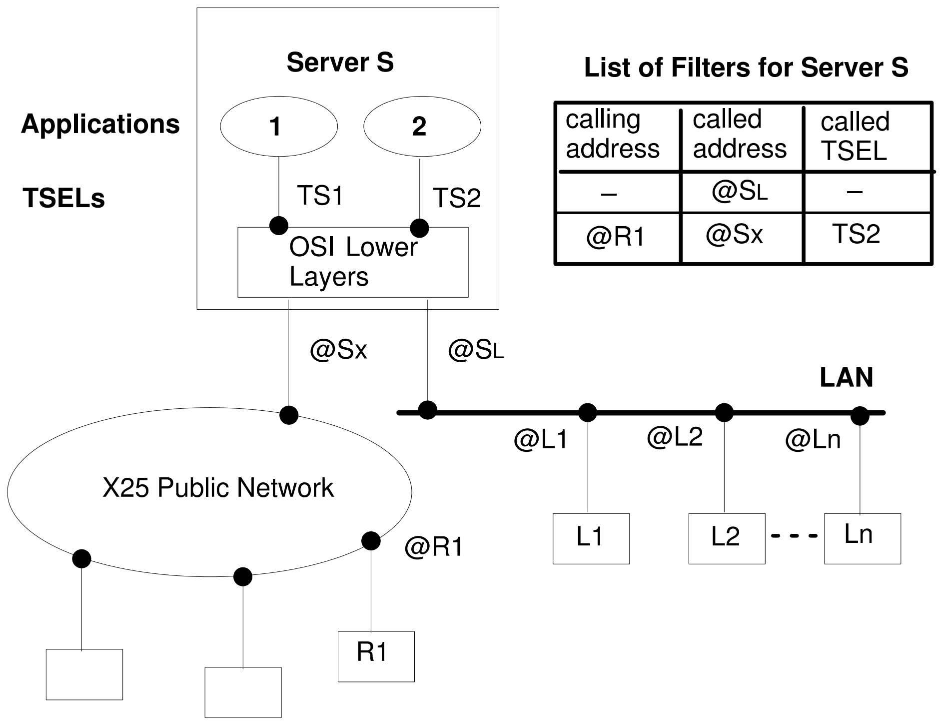


Figure 28. Secured System Control Points

The two control points check calling and called network addresses. The transport control point also checks the called TSEL. Each control point can be separately activated or de–activated. The address parameters are either NSAP address, LAN MAC address, X.25 DTE address, PVC name, or '–' (to mean any address).

Security Example

The aim is to ensure that the stations **L** connected on **LAN** can use any application while R1 is only able to connect to Application 2. The principle of operation is shown in Figure 29.



List of Filters for Server S

calling address	called address	called TSEL
–	@SL	–
@R1	@Sx	TS2

Figure 29. Secured System Example

Chapter 6. Supporting Your Network Topology

This chapter is dedicated to administrators who are already familiar with OSI networking. It provides detailed information concerning the use of LAN and X.25 networks together with network interconnection.

Supporting Your Network Topology Overview

An operational network is usually a set of interconnected networks with different characteristics. The OSI Stack supports LAN networks (Ethernet, Token Ring, FDDI) and X25 networks. This section is for administrators already familiar with OSI networking and describes how to make the best use of these networks and how to interconnect them. You can find more information in:

- Use of LAN Subnetworks, on page 6-2.
- Dynamic Versus Static Routing, on page 6-4.
- Use of X25 Subnetworks, on page 6-5.
- Subnetwork Interconnections, on page 6-11.

Use of LAN Subnetworks

One of the main advantages of LAN for ease of use, is the support of broadcasting facilities which allow dynamic routing protocols to be developed. The aim of these protocols is to automatically solve the CLNP routing issues. The OSI Stack supports the ES–IS protocol which is describe here.

ES–IS Protocol Description

ES–IS protocol on LAN is shown in Figure 30.

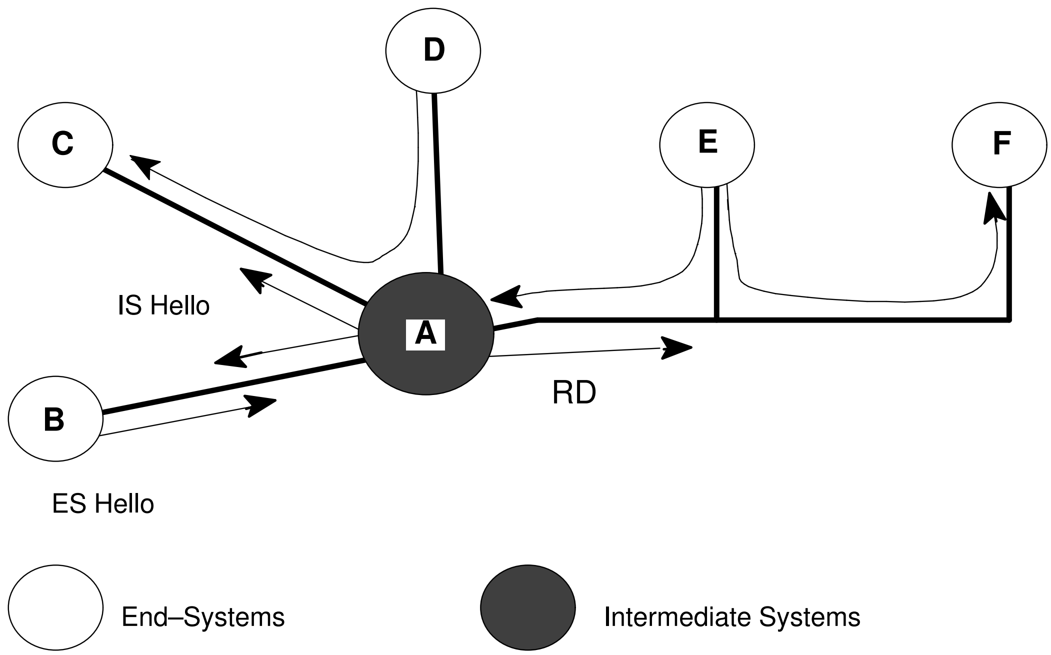


Figure 30. ES–IS Protocol on LAN.

The protocol mechanism is as follows:

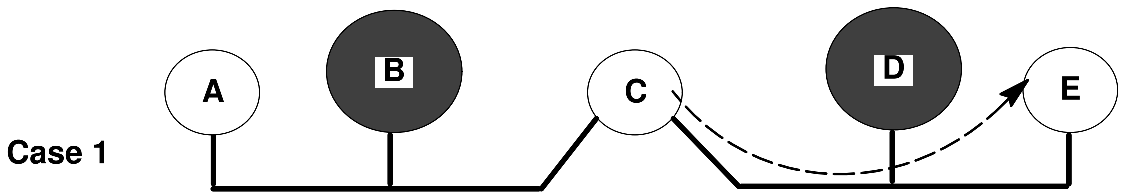
- the Intermediate System (IS) broadcasts periodically IS Hello (ISH) PDUs to the End Systems (ES) to be known by these systems. The ES broadcasts periodically ES Hello (ESH) PDUs with lists of local NSAPs. The IS registers all the NSAPs of connected ES.
- Routing is made by IS. When an ES must send a PDU to an unknown recipient, it sends the PDU to one of the known IS. The IS will ensure the PDU routing. If no IS is known, the PDU is multicast to all ES, waiting for a reply from the destination ES.
- Redirected (RD) PDUs may be generated by an IS to allow two ES using the same physical network to communicate directly.

When operating on a single LAN, it is not necessary to have an IS configured. ES will send the PDU to an unknown recipient using broadcast, waiting for an answer from the destination ES. Further PDUs will be sent in point–to–point mode. As soon as networks are interconnected, it is mandatory to configure the relay system as an IS.

ES–IS Limitations

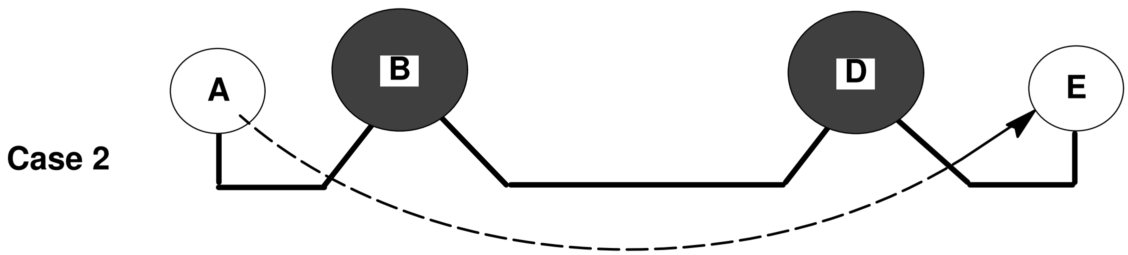
The ES–IS is limited to LAN subnetworks. There is no information exchange between IS. It is not possible to interconnect more than 2 networks serially using only dynamic addressing.

ES–IS limitations are shown in Figure 31.

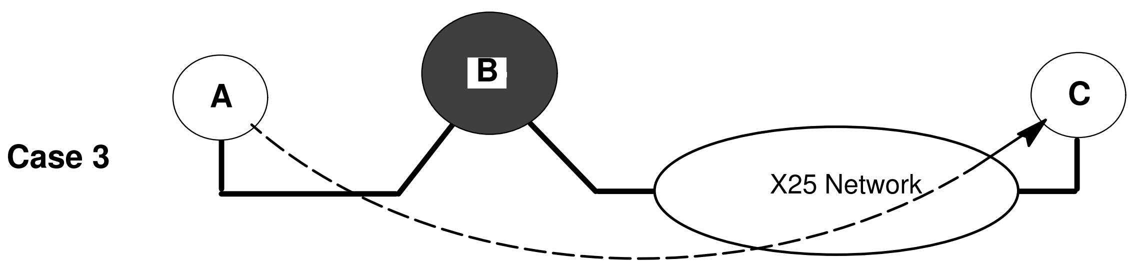


Case 1: C knows a single IS. When it is B, it is not possible for C to reach E.

In Case 1, the problem is solved if C is declared as an IS and B and D as ES.



Case 2: A knows B, but B does not know E because they are not on the same physical network. E is known only by D.



Case 3: A knows B, but B does not know C because ES–IS protocol is limited to LANs.

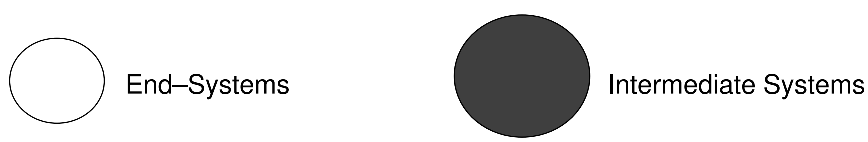


Figure 31. ES–IS Limitations.

Dynamic Versus Static Routing

Dynamic Routing

Dynamic routing protocols remove the need for the administrator to configure routing information. The single configuration action which is necessary is to define whether each station is ES, IS or both. The protocols then automatically detect stations with new NSAPs. Full CLNP profiles must be used to benefit from these protocols. Support is limited to LANs only.

Static Routing

Static routing requires that all routing information be configured by the administrator. It has the advantage that it allows the limitations of dynamic routing protocols to be by–passed on both LAN and X25 networks. It also allows an optimized routing plan to be defined.

Static routing may work in conjunction with dynamic routing. Routes defined statically have a higher priority than dynamic routes: in case of conflict they are then selected first.

For more information concerning static addressing, see Configuring, Starting & Stopping the OSI Stack, on page 8-1.

Use of X.25 Subnetworks

Support of X.25 80, 84, 88

The OSI Stack does not have the knowledge of the X25 version available through the X25 communications adapters used. It does not perform any control, but may introduce limitations which depends on the X25 profile used:

Profile Name	X25–80	X25–84	X25–88
Full CLNP	y	y	y
CONS	n	y	y
X25	y	y	y

The profile CONS does not support X25–80 because it uses addressing facilities to convey NSAPs. These facilities have been introduced in X25–84.

It is possible to configure applications to allow them to send X25 facilities and call user data (CUD). The administrators and developers must take care that these facilities are compatible with the X25 version supported by the communications adapter and network used, else the connection will be refused by the X25 driver or by the X25 network.

Facility Management

Here is described how it is possible to manage X25 facilities with the OSI. Limitations introduced by the communications adapter, or the X25 network used are not taken into account.

To get information about facilities supported by the X25 communications adapters please refer to the adapter documentation. When an X25 network is used, check also the characteristics of the X25 attachment.

Facility Processing

This paragraph describes how facilities are processed by the OSI Stack when operating over COTP, COSP or ACSE layers. Developers interested in X25.3 interface must refer to MAX3 documentation, see MAX25.3 Programmer's Guide in Bibliography, on page B-1.

The general mechanism is:

- the connection originator proposes facilities,
- the stack may add some facilities,
- the connection recipient may be notified of the facility negotiated,
- the connection originator does not know the result of the facility negotiation between local and remote X25 provider.

With the considered interfaces, it is not possible to negotiate the X25 facilities: above COTP layer, the attempt of transport connection establishment requires to open an X25 virtual circuit. The circuit is always established with the proposed facilities. Then the connection recipient can only close the transport, session or ACSE connection if it does not agree with the negotiated facilities.

The processing of facilities depends on the profile used as described in the next table. The behavior is the same for all the APIs.

Profile Name	Connection Originator	Connection Recipient
Full CLNP	Proposed facilities are ignored CUG is transmitted when configured in the RIB. Priority is transmitted when the priority parameter is specified.	no facilities
CONS	Proposed facilities are transmitted. COTP adds calling and called addresses extension, priority, CUG (when configured in the RIB) if these facilities are not specified in the proposed facilities.	all facilities are received, including facilities added by COTP.
X25	Proposed facilities are transmitted.	all facilities are received.

The Stack relays may also generate facilities for the outgoing connection. The MSDSG and CLNP relay provide the CUG stored in RIB (see CONS Profile). The transport relay gets these facilities from the profile database.

Furthermore, it is possible to configure the stack in order to add facilities (if they are not already specified) in the X25 CALL REQUEST and CALL CONFIRM packets on a per line basis. This mechanism can be used by all the stack APIs and X25.3 access method. These facilities are listed in the next table.

	Call Request	**Call Confirm**
negotiation of packet size	X	
negotiation of window size	X	
reverse charging request	X	
negotiation of throughput class	X	
negotiation of minimum throughput class	X	
fast select	X	
transit delay	X	
end to end transit delay	X	X

Application Visible Facilities

Application visible facilities are facilities which can be defined on a per call basis. The next table list these facilities, and indicate which are visible. The facility name is the same the one used in X25–88 standards.

	COTP/COSP/ACSE	
Facility	**Connection Request**	**Connection Indication**
X25 facilities		
flow control parameter negotiation (packet size)	X	X
flow control parameter negotiation (window size)	X	X
throughput class negotiation	X	X
closed user group selection (basic format)	X	X
closed user group selection (extended format)	X	X
closed user group with outgoing access selection (basic format)	X	X
closed user group with outgoing access selection (extended format)	X	X
bilateral closed user group selection	X	X
fast select	X	X
network user identification selection		
charging information (requesting service)	X	X
charging information (indicating monetary unit)		
charging information (indicating segment count)		
charging information (indicating call duration)		
RPOA selection (basic format)	X	
RPOA selection (extended format)	X	
call deflection selection		
call redirection or call deflection notification		X
called line address modified notification		
transit delay selection and indication	X	X
CCITT facilities		
calling address extension	X	X
called address extension	X	X
minimum throughput class negotiation	X	X
end to end transit delay	X	X
priority	X	X
protection	X	X
expedited data negotiation	X	X
marker	X	X

Developers must refer to the API documentation to know how to send or receive facilities.

Call User Data & SPI

Applications are able to specify X25 CUD. The general mechanism is:

- the connection originator provides CUD,
- the connection recipient may receive CUD,
- incoming CUD are processed by the OSI Stack.

The processing of CUD depends on the profile used as described in the next table. The behavior is the same for all the APIs.

	Connection Originator	**Connection Recipient**
Full CLNP	provided CUD are ignored CUD 0x81 (1 byte length) is sent	no CUD received
CONS	provided CUD are transmitted	CUD are received
X25	provided CUD are transmitted	CUD are received

The received CUD are used by the OSI Stack to dispatch incoming calls between CLNP, COTP and applications on top of X25.3. The dispatching of incoming calls is configurable according to the CUD. It is possible to configure SPI values (between 1 and 4 bytes), and to switch according to these values between COTP, CLNP and applications on top of the X25.3 interface. It is also possible to define recipients for unknown CUD and for missing CUD.

It is not recommended to define CUD for applications on top of COTP, COSP and ACSE/Presentation except for inter–operability purposes.

The transport relay may also provide CUD (which are stored in the profile database) for the route under consideration.

Line Control

Lines are opened during stack loading. They are close before unloading. There is no way to close and reopen lines during stack activity, nor to defer line opening.

SVC/PVC

The OSI Stack supports both SVCs and PVCs, but with some restrictions for PVCs.

	SVCs	PVCs
Full CLNP	yes	no
CONS	yes	See note
X25	yes	yes

Note: A connection originator may use PVCs with CONS addressing parameters, but the calling and called NSAPs are not transmitted to the remote station: the connection is therefore not established with a CONS profile and the recipient will have the visibility of an X.25 profile.

PVCs are identified by a name. The relation between the name and the circuit number must be configured by the administrator. The configuration action also defines if the PVCs are used by COTP or applications on top of X25.3.

Addressing

As soon as full CLNP or CONS profiles are used over X25, routing information must be configured statically.

For more information concerning static routing configuration, see Configuring, Starting & Stopping the OSI Stack, on page 8-1.

Subnetwork Interconnections

Relays

The OSI Stack Contains several relays which allow supported networks to be interconnected. The relay to be used depends upon the connection profile used. A common requirement is to use NSAP addressing. Three supported relays, CLNP, MSDSG and Transport are described here.

CLNP Relay

The CLNP relay requires the use a full CLNP profile on both sides of the relay. It is able to relay between any couple of LAN or X25 subnetworks.

Only NPDUs not destined for the local station can be relayed by CLNP relays.

If the called NSAP is not a local one, CLNP checks the information on 'how to route an NPDU'. If the information is not available in the static RIB (i.e. the route is unknown), or if the PDU must be routed using CLNP relay, the CLNP relay function is activated (MSDSG is not used) and the dynamic routing protocols may be activated (unknown route).

On LANs, the relay works in conjunction with ES–IS protocol.

The use of this relay is fully transparent for the protocols above CLNP. It is possible to relay both connection–oriented and connectionless transport protocol. There is no addressing conflict with local applications because routed data is not destined for the the local station.

MSDSG Relay

The MSDSG relay is a transport relay, but it behaves like a CLNP relay. The relaying criteria is the called NSAP; calling and called NSAPs are left unchanged by the relay. Only connection–oriented transport protocol can be relayed. It is as transparent as possible for the transport protocol, even if it is able to perform transport class protocol conversion. It tries to maintain the same parameters on the outgoing connections as were present on the incoming connection. A transport connection is established on each side of the relay, but the transport user has the visibility of an end–to–end connection. The relay performs TSDU relaying.

The outgoing transport connection established by the relay has the following characteristics:

- propose class 4 and alternative class 0 (negotiation to lower classes possible).
- same value for priority parameter as received in the incoming connection.
- flow control proposed.
- same TPDU size as the received size.
- use of extended format and checksum as specified in the incoming connection if received class is 4, else use of these options is not proposed.
- same credit as received in the incoming connection. If the incoming connection proposes class 0, a credit 1 is proposed on the outgoing connection.

The relay does not provide specific CUD or X25 facilities. To know which facilities may be added by the transport provider, see Use of X25 Subnetworks, on page 6-5.

The MSDSG can operate regardless of the system type (ES, IS, or both).

The incoming connection is established only if the outgoing connection has been established successfully.

The MSDSG relay cannot relay between any couple of profiles using NSAP addressing. For example, the outgoing connection on X.25 cannot use a full CLNP profile.

Use of the MSDSG relay is transparent for applications above the transport layer; they need no knowledge of its presence.

As for the CLNP relay, there is no addressing conflict with local applications because data relayed by MSDSG is not destined for the local system.

Transport Relay

This relay relays only the connection–oriented transport protocol (COTP) and uses a transport connection on each side of the relay, like MSDSG. This is the unique common point between the two relays.

From an addressing point of view, the transport relay is not transparent. Applications must connect to the relay system and so do not provide the address of the final system. For incoming connections, the relay behaves like an application. It is therefore recommended that normal applications avoid waiting on an address that the relay is able to route. In case of conflict (for example, because of the use of wildcard conventions) a rule defines how the incoming call is handled, see Addressing, on page 3-1.

The relay does not try to reuse parameters of the incoming connections. It uses the profile defined by the administrator, and if none is found, parameters are generated using default rules.

When the outgoing connection is established, the relay confirms the incoming connection, using the transport class specified in the selected route, if this class is lower than the required class. Care must be taken when configuring another value than 4: because of transport class negotiation rules, some values may lead to a disconnection.

Here is an example to avoid:

Configured class = 0

Incoming connection with preferred class 4, alternative class 2

The relay responds with class 0, which is not a valid response.

Chapter 7. Installing the OSI Stack

Installing the OSI Stack Overview

This section addresses an administrator who must install an OSI Stack. He should be familiar with AIX installation tools.

Installation procedures are detailed in:

- Understanding OSI Stack Packaging, see page 7-2.
- Prerequisite Software, see page 7-4.
- Installing the OSI Stack, see page 7-6.
- After Installing the OSI Stack, see page 7-7.
- License Control – iFOR/LS, see page 7-9.
- De–installation Procedure, see page 7-11.

Installation Terminology

The terminology and organization of packaging of all software for AIX Version 4.1 is as follows:

- Software is delivered in LPPs; the term **Licensed Program** is sometimes used instead of LPP.
- An intermediate level called the **package** allows the grouping of several OPPs (or options) in a set. OPPs are sometimes called **filesets**.

Understanding OSI Stack Packaging

Software packaging and installation details are to be found in the *AIX Installation Guide* see Bibliography, on page B-1.

The stack is composed of three packages, each of which has one or more OPP:

- **Stack Framework**, which contains general software and configuration tools. It is mandatory in order to use Low Layers or High Layers.
- **Stack Low Layers**, which contains the OSI layers 1 to 4.
- **Stack High Layers**, which contains the OSI layers 5 to 7.

For more information on the choice of High of Low layers, see Planning Your Installation, on page 5-1.

Stack Framework

The stack framework LPP (**osi_frame**). This LPP is a prerequisite for the **osi_low** and **osi_high** LPPs. The framework LPP contains the following OPPs, which are co-requisites (must be installed or removed together):

- **osi_frame.rte**, contains common services for Stack layers (MAD environment), daemons, configurators, and SMIT menus.
- **osi_frame.odit**, contains ODIT commands, tools, and SMIT menus.

As soon as one of these filesets is installed, the other is also automatically installed.

Stack Low Layers

The stack low layers LPP (**osi_low**). The MAX3 access method, previously part of the TPAD-HPAD product, is now part of this LPP. This LPP is optional if TCP/IP is used for the lower layers (possibly with NetShare). Finally, the lower OSI layers are delivered in a single STREAMS module. The LPP contains the following OPPs:

- **osi_low.rte**, contains the low stack layers (1 to 4) in a single STREAMS module.

  Optional Fileset

- **osi_low.max3**, contains include files and example programs for the the X25.3 access method.

Stack High Layers

The stack high layers LPP (**osi_high**). The LPP contains streams drivers that are installed in the **/usr/lib/drivers/pse/osi** directory and run time parts of the Session API and the XAP API. There is one option for the trace stanzas of each of the following APIs: XAP, ROSE, and Session. There is also one option that contains the CIS configuration menus for XAP. These options only have to be installed if the corresponding APIs are used. Other high level stack APIs and their trace stanzas are part of separate LPPs.

This LPP is optional if OSI upper layers are not required.

The **osi_high** LPP contains the following OPPs:

- **osi_high.rte**, contains the upper stack layers (5 to 7).

  Optional Filesets

- **osi_high.smit.xap**, contains the CIS configuration menus for XAP.
- **osi_high.trc.xap**, contains trace services for XAP.
- **osi_high.trc.apises**, contains trace stanzas for the session API.
- **osi_high.trc.rose**, contains trace stanzas for the ROSE API.

Prerequisite Software

The following table describes the prerequisites for each fileset.

Package you are installing	**Prerequisite software** (Install before or **with** option you are installing)	
osisnap	none	
osi_frame	osisnap	no prerequisite on AIX packages
osi_low	osi_frame	
osi_high	cis osi_frame & either osi_low or NetShare	cis required only by osi_high.smit.xap option

Migration Installation Instructions

Migration from an earlier Stack and/or version of AIX or re–installation of the same version is supported. The information configured by the administrator is conserved.

Refer to the *OSI Communications Porting Guide* , see Bibliography, on page B-1 for more information about the migration process, versions which can be migrated and differences with previous versions.

Installing the OSI Stack

All LPPs are installed using AIX commands and tools (SMIT, **installp**, VSM interface AIX desktop). The product can be installed on both standalone and diskless stations, but the installation procedure is different for each type of station.

CAUTION:
It is mandatory to de–activate the OSI Stack before installing a new version or attempting a re–installation of the current version. If the Stack is not de–activated, the installation is refused.

OSI Lower Layers

When installing the OSI Stack lower layers, all the prerequisite software can be installed automatically. The installation of the lower layers must normally be selected by the administrator before installing any other OSI communications product (when OSI lower layers are used). The minimum fileset to install is *osi_low.rte*.

OSI Upper Layers

Once the lower layers have been installed, the administrator may directly install the applications. The application installation will trigger the installation of the Upper Layers (when they are required) if the automatic installation of prerequisites is selected.

Note: The installation of Upper Layers does not trigger the installation of lower layers (because of the use of the NetShare or osi_low package).

When using XAP, Session or ROSE API, it is recommended that the Stack Upper Layers are installed before installing the API packages.

When installing OSI Upper Layers, the minimum fileset to install is *osi_high.rte*.

CAUTION:
For the processes of prerequisite installation and de–activation of the Stack, it is recommended to select the automatic option.

Optional Installation of Diagnostic Tools

The OSI diagnostic tools, described in the *OSI Diagnostic Interactive Toolkit (ODIT) User's Guide* see Bibliography, on page B-1, are automatically installed as soon as the OSI upper or lower layers are installed.

It is possible to install trace grammar and ODIT menus for the upper layers APIs used by Bull products for very difficult problem analysis. The minimum fileset to install is *osi_high.rte*.

After Installing the OSI Stack

This paragraph explains the results of the OSI Stack installation.

OSI File Location

OSI Driver *\/usr/lib/drivers/osi*

This file contains the framework driver and reference files.

OSI Stream Drivers */usr/lib/drivers/pse/osi*

This directory contains all the stream drivers corresponding to the protocol modules.

Commands and Shell */usr/bin*

The shell scripts and most of the commands use the *osi* prefix.

Maintenance Tools */usr/bin*

There are several files providing error and trace management facilities.

OSI Configuration Data Base */etc/objrepos/osi*

The files in this directory store the configuration information (default and customized values).

System File Changes

To be able to use the OSI services, actions are automatically performed at boot time, using the following definitions, added during the installation.

/etc/inittab

The following lines are added to the *inittab* file:

osidaemon : 2 : wait : /usr/bin/osiboot >/dev/console 2>&1

 # Start Stack OSI

See Starting & Restarting the OSI Stack, on page 8-36.

/usr/lib/objrepos/PdDv

The OSI driver is defined in the Pre–define Device Class with the attributes shown below:

```
PdDv:
        type = "OSI"
        class = "osistack"
        subclass = "commo"
        prefix = "mad"
        devid = " "
        base = 0
        has_vpd = 0
        detectable = 0
        chgstatus = 1
        bus_ext = 0
        fru = 0
        led = 0x599
        setno = 1
        msgno = 1
        catalog = "osistack.cat"
        DvDr = "/etc/osistack"
        Define = "/etc/methods/defmad"
        Configure = "/etc/methods/cfgmad"
        Change = " "
        Unconfigure = "/etc/methods/ucfgmad"
        Undefine = "/etc/methods/udefmad"
        Start = " "
        Stop = " "
        inventory_only = 0
        uniquetype = "osistack/commo/OSI"
```

/usr/lib/objrepos/sm_*

Installation of SMIT menus.

/etc/trcfmt

Installation of trace grammar for OSI Stack and optionally XAP, ROSE and Session API.

/var/adm/ras/errtmplt

Installation of template for AIX error login service.

Daemons

The following daemons are used by the OSI Stack:

- *osinetlsd*
- *osilinkd*
- *osiribd*
- *dat_x25.*

License Control – iFOR/LS

License Management

This product uses iFOR/LS encrypted license keys for license management. It supports the 'Nodelocked' license type only. License status is validated only when the product is used, thus permitting installation and configuration without need of the license key.
Refer to the iFOR/LS Installation Notice and Password Order Form delivered with your Communications Product.
Refer to the SRB file for details on how the product uses the license key.

Applicable documents are listed in the Bibliography, on page B-1.

OSI Layer Licenses

The licenses are **nodelocked** licences: the licence contains the reference of the machine for which it has been ordered, and it is not possible to install the same license on several nodes, or to change licenses between nodes.

Upper Layers, Lower Layers and X25.3 Access Method can be separately enabled or disabled according to the license installed.

Telecommunications features, grouped with the OSI Stack, are called Communications Bundles and can be managed in a similar manner.

The following table decribes the relationship between license and enabling or disabling of components.

License	Upper Layers	Lower Layers	X25.3 A M
Bundle (Basic communications)	X	X	X
Layers 5 to 7	X		
Layers 1 to 4		X	
X25.3 Access Method			X

The license may have a limited duration. A hard stop policy is implemented:

- for upper layers, all connections will be closed, and new connections will be refused if the license is missing, wrong or out of date,
- for lower layers, all connections will be closed, new connections will be refused, and relay features will be disabled if the lower layer license is missing, wrong or out of date.

The X25.3 Access Method, which is embedded in the Lower Layers has its own Licence.

OSI framework and osisnap are license–free packages.

License Validation

The Stack licenses are checked during stack activation. The Stack is activated as soon as the license enabling OSI Lower Layers, OSI Upper Layers or X25.3 Access Method is detected. However, the stack layers, loaded in kernel memory, depend on the packages installed, regardless of the license.

These licenses are periodically checked, every 5 minutes, by the *osinetlsd* daemon. A new license is taken into account at the next checking period, or at the next activation.

The Stack commands do not check the license availability: it is possible to configure, activate and de–activate the stack even if there is no license.

Troubleshooting tools will help you to determine if the licenses are well installed and take into account. The OSI Diagnostic Interactive Toolkit (ODIT) User's Guide gives details, see Bibliography, on page B-1.

De–installation Procedures

CAUTION:
Before attempting de–installation, ensure that the OSI Stack has been stopped and unloaded, see Configuring, Starting & Stopping the OSI Stack, on page 8-1.

When de–installation is performed, all configurations, defined by the administrator, are lost.

Chapter 8. Configuring, Starting & Stopping OSI Stack

Describes basic operations with OSI Stack and Applications.

Configuring, Starting & Stopping the OSI Stack: Overview

This section helps you to understand which information must be configured, the order, and why. To use the OSI Stack follow the main steps featured in Figure 32.

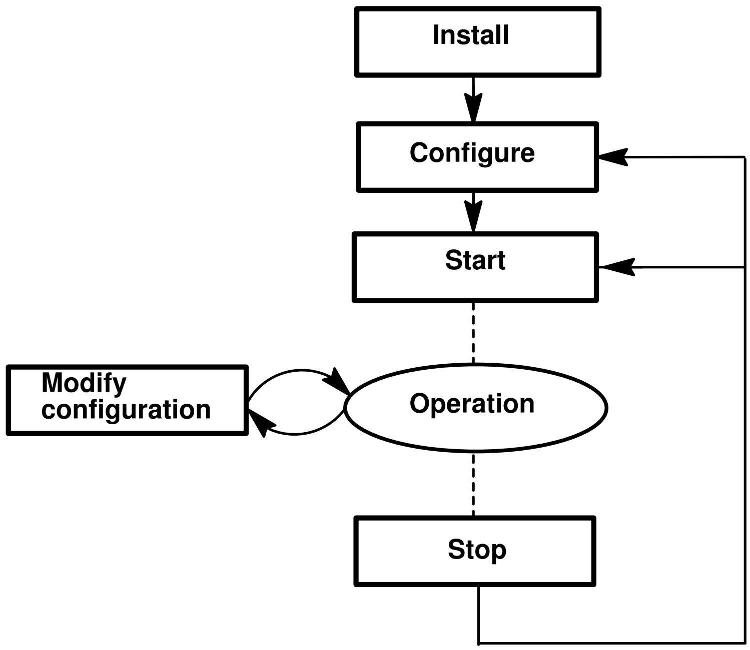


Figure 32. Using OSI Stack – Functional Flow

You can find more information in:

- Configuring the OSI Stack, on page 8-3.
- Understanding Default Values, on page 8-8.
- Understanding Default Configuration, on page 8-10.
- Understanding Network Access Configuration, on page 8-11.
- Understanding PVC Configuration, on page 8-13.
- Understanding LSAP Configuration, on page 8-30.
- Understanding Dynamic Routing, on page 8-15.
- Understanding Static RIB Configuration, on page 8-16.
- Understanding Transport Relay Configuration, on page 8-23.
- Understanding Profile and Route Configuration, on page 8-27.
- Understanding Security Filter Configuration, on page 8-29.
- Understanding Local NSAP Configuration, on page 8-14.
- Understanding Netbios Configuration, on page 8-31.
- Understanding Stack Tuning, on page 8-32.
- Operating System and Driver Configuration, on page 8-34.

Configuring the OSI Stack

The configuration phase is mandatory before using OSI Stack services. The aim of this configuration is to:

- define local addressing information (communications adapters, NSAPs, ...),
- define information to reach remote stations,
- customize or tune the Stack,
- manage communications security.

How to Configure

The OSI Stack can be configured as described in this manual.

Configuration commands, see page 10-1, are also integrated in SMIT, see OSI Stack Configurator, on page 9-1. The commands ensure the consistency of the configured parameters. The configured information is stored in the Object Database Manager (ODM) database using a private format. The ODM database must not be modified using ODM commands.

Another way to configure the OSI Stack is to use the simplified configurator which is delivered with **Bundle** solutions. This restricted configurator provides the visibility of a subset of stack functions which are sufficient for the Bundle solutions. For more information, see *BASIC-COM and GCOS-COM Installation Guide*, Bibliography on page B-1.

Access Rights

Two levels of access are defined:

1. All users may display information from the OSI configuration data base or may obtain information using the dynamic configuration services.
2. Only a system administrator with super user authority can modify the OSI configuration data base, generate configurations, activate and de–activate configurations, and modify information with the dynamic configuration services.

What to Configure

A stack configuration contains objects, and parameters. It is possible to list, add, remove, and change objects. Parameters refer to values and can only be changed. Objects describe entities which may have several instances in the stack. Most of the information can be changed dynamically without disturbing the OSI Stack service. But some information can be changed only when the stack is not active.

The two following tables list the objects and parameters, and describe what can be changed dynamically. (Y = Yes, N = No, X = Y or N, N.S.= Not Supported). The column "modify" shows which parameters are optional (O) and which ones are reserved for expert administrators (R).

Note: Static parameters must be configured before starting the OSI Stack to avoid disturbing stack users.

Table of Object Descriptions

	Description	**Add**	**Remove**	**Change**	**List**
configuration	set of objects and parameters	Y	Y	Y	Y
communica-tions adapters	network attachment used by the OSI Stack	Y	Y	N	Y
local NSAPs	local NSAPs	Y	Y	N	Y
subnetworks	logical identification of a communications adapter	Y	Y	Y	Y
RIB entries	network routes for static routing	Y	Y	Y	Y
PVCs	logical identification of PVCs	N	N	N	Y
X25 facilities	facilities added by the stack	N	N	N	Y
relay route	routing information for transport relay	Y	Y	N.S.	Y
profile	profile definition	Y	Y	N.S.	Y
routes	remote address with an attached profile	Y	Y	N.S.	Y
security filters	information to filter incoming connection	Y	Y	N.S.	Y

A configuration itself is an object: it is possible to define several configurations. This feature should be used to make a backup of a configuration before attempting any changes. Each configuration is identified by a name, and has a state which is either:

- **idle:** the configuration exists in the ODM data base,
- **selected:** list, add, remove and change operation applies to the objects and parameters of this configuration,
- **loaded:** the configuration corresponds to the stack currently loaded.

When the configuration is loaded and selected, any change to a dynamic parameter or object is performed in both ODM database and loaded stack. Modifications to static parameter or objects are refused.

There is always, at most, one selected configuration and one loaded configuration (usually the same).

Table of Parameter Descriptions

A new configuration can be created by copying an existing one or with the default values described in the next table.

Set	Name	Description	Default Parameters	Modify	Change
Presentation Layer	COPP connections	maximum number of COPP connections	64	O	Y
Session Layer	COSP connections	maximum number of simultaneous COSP connections.	64	O	Y
	COSP connection timer	QOS connection establishment delay	160 s	R	Y
	COSP disconnection timer	QOS connection disconnection delay	40 s	R	Y
	COSP indication timer	incoming retention time when no listener available (implementation specific)	5 s	R	N
Transport Layer	COTP connections	maximum number of simultaneous COTP connections.	64	O	Y
	COTP multiplexer	number of COTP connections multiplexed on a virtual circuit		O	N
	COTP re–transmission timer (LAN)	delay between retransmissions (LAN)	3 s	R	Y
	COTP re–transmission timer (WAN)	delay between retransmissions (WAN)	15 s	R	Y
	COTP retransmissions (LAN)	maximum number of transport retransmissions	10 s	R	Y
	COTP retransmissions (WAN)	maximum number of transport retransmissions	10 s	R	Y
CLNP / ES–IS	ES–IS protocol state	ES–IS protocol can be enabled or disabled	enable	O	Y
	system type	defines the station type for ES–IS protocol: ES, IS or both.	ES	O	N
	ES timer	delay between ESH	180 s	R	N
	IS timer	delay between ISH	180 s	R	N
LLC	LSAPs for null CLNP profile		0x20	R	N
X.25	SPI	dispatching of SVCs according to X25 CUD		R	N

Set	Name	Description	Default Parameters	Modify	Change
Global parameters for lower layers	Buffer memory size	limitation for memory allocation in lower layers		R	Y
Netbios	multicast NSAP			R	N
	multicast LSAP			R	N
	subnetwork			R	N

For details on the possible values, ranges and defaults, see OSI Stack Configurator, on page 9-1 for the SMIT interface and configuration commands, on page 10-1. Configuration & Limits, on page 4-1 lists all the configurable parameters and/or default values. All the optional parameters are grouped in a single SMIT screen.

The relationship between objects, parameters and Stack is shown in Figure 33.

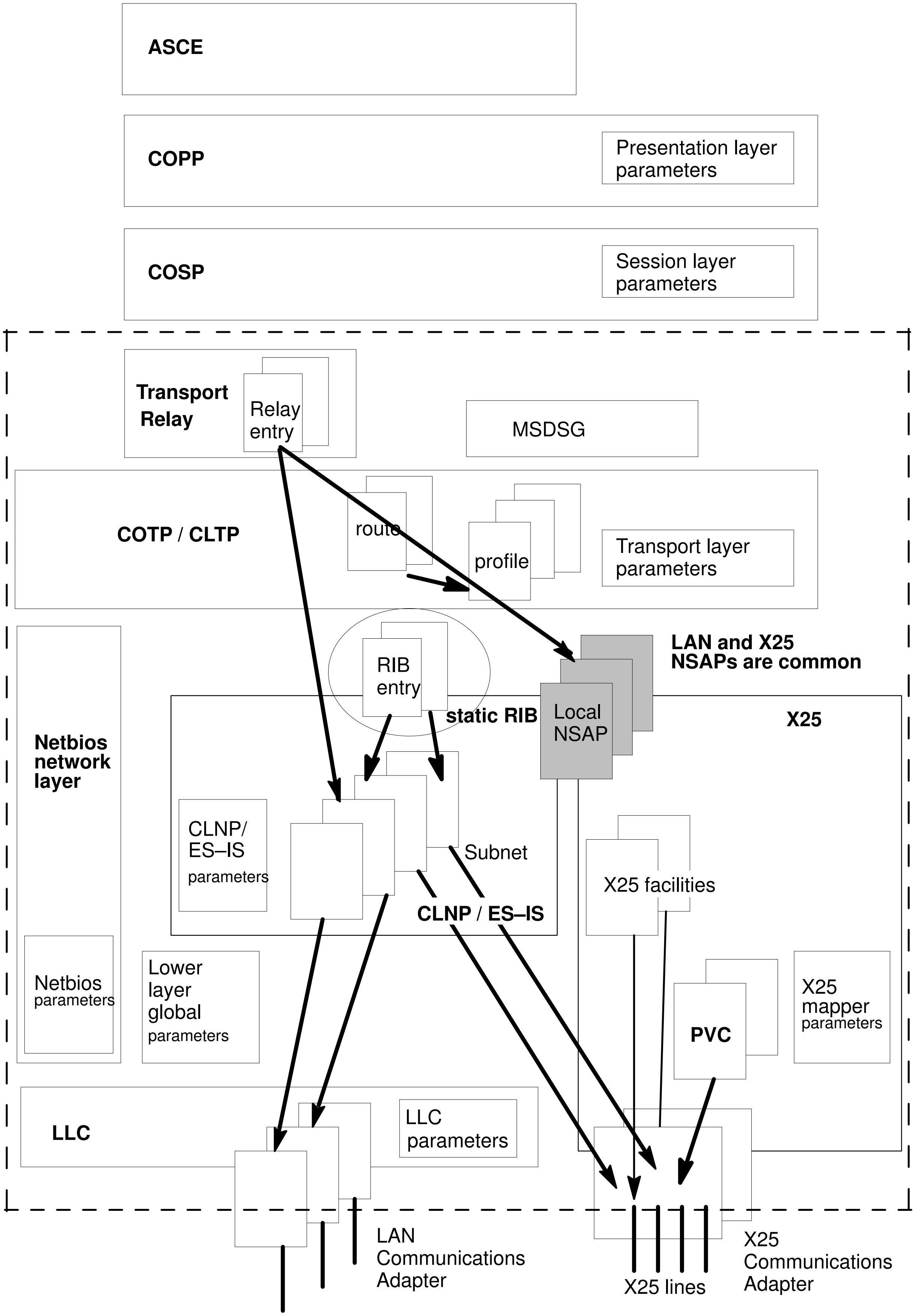
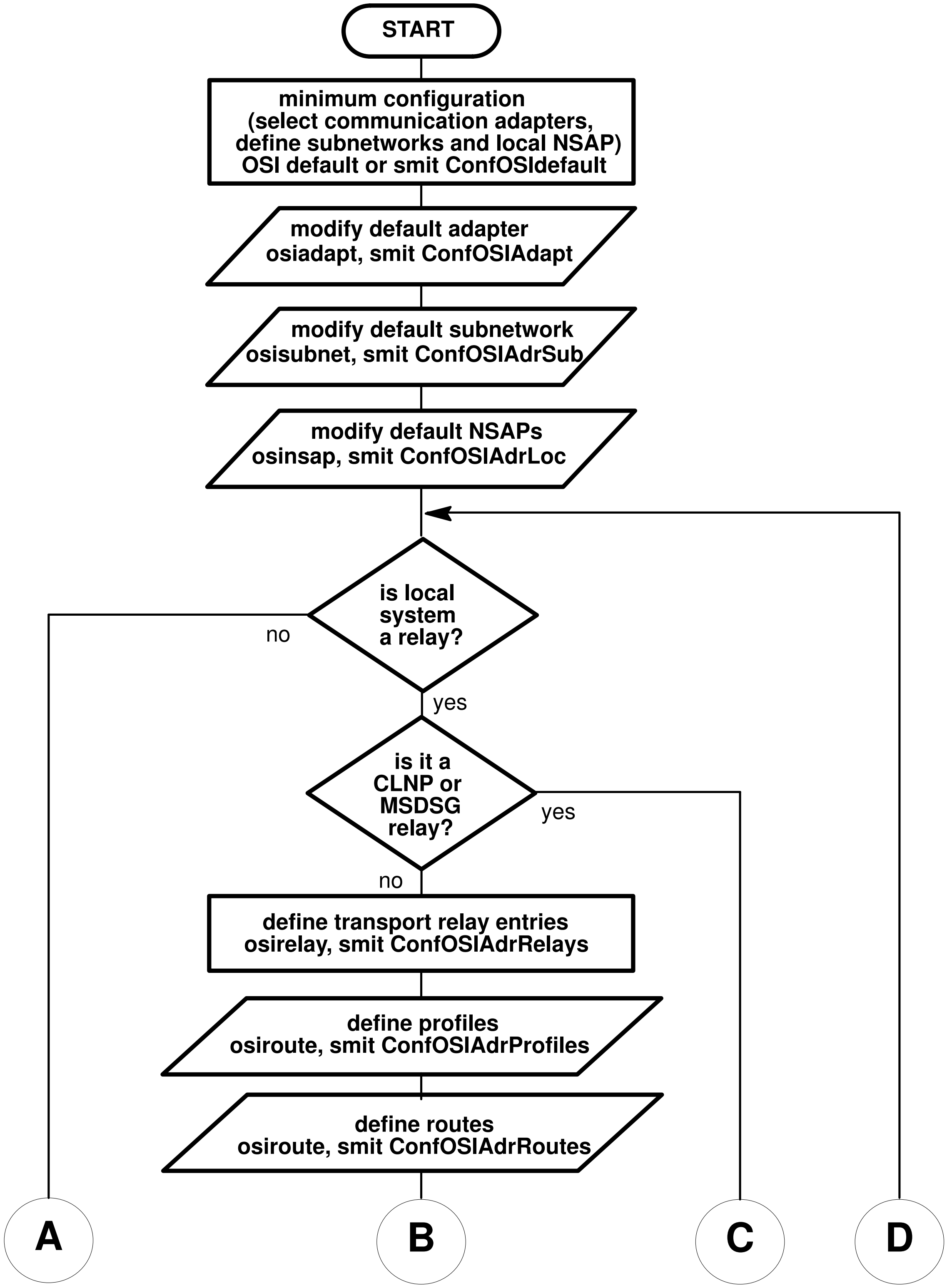


Figure 33. Relationship Between Objects, Parameters & Stack

Configuration Flow Chart

Figure 34 proposes a decision tree to help you to configure the right information in the right order for the first configuration of the Stack. The actions and choices are discussed in the next paragraphs. Further configurations re–use sub–parts of the whole tree. First configure communications adapters, then do:



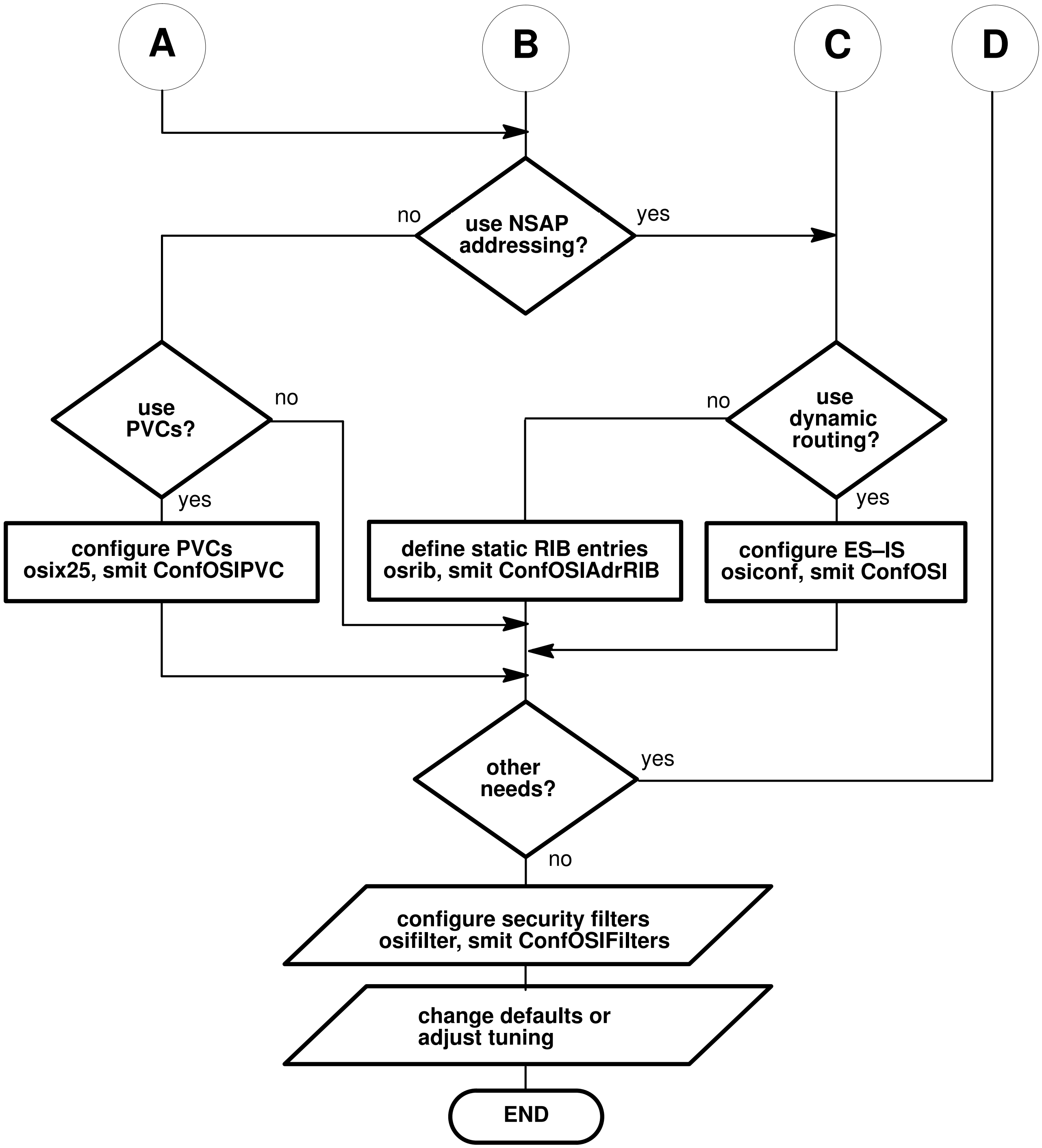


Figure 34. Creation of a Configuration – Flow Chart

Understanding the Use of Minimum Configuration

To simplify the configuration phase, a SMIT menu **smit ConfOSIdefault**, on page 9-36 and a command line **OSIdefault**, on page 10-11 automatically creates a configuration with default values.

The created configuration allows all the supported communications adapters to be used. Subnetworks are configured for each communications adapter. Default local NSAPs are generated. The ES–IS protocol is enabled and the stack is configured to be an End System. All default values are used.

Two default NSAPs are built:

- the first is built on the station name (returned by the **uname –n** command). The returned ascii string is coded in hexadecimal, prefixed by 0x49 and suffixed by 0x01.
- the second is based on the station–unique ID (also used to define nodelocked licenses). The **uname –n** command returns the value 00<unique ID>00. ID is in 4–bytes. The NSAP used the unique ID, prefixed by 0x49 and suffixed by 0x01.

CAUTION:

1. These NSAPs should be unique, but as local NSAPs are configurable, there is no guarantee that the same NSAPs will not exist on another station.
2. It is not possible to activate OSIdefault configuration twice. The configuration must be removed and then re–created.
3. When communications adapters are added, removed or moved after OSIdefault has created the default configuration they are not taken into account automatically and must be re–configured by the administrator.

Example of Generated NSAPS:

```
result of ”uname –n –m” command:
rs3 000085573500
Hexadecimal coding for rs3 is 0x727333.
result of ”ls_targetid” command:
                PREFERRED iFOR/LS Target ID (CPU Planar ID)
                ------------------------------------------
                        855735

                ALTERNATE Target ID (Link Level Address)
                --------------------------------------------
                        78B7010D

NSAPs generated (osinsap –l) :
0x4972733301            LOCAL
0x490085573501          LOCAL
```

Understanding Network Access Configuration

Network access is established via a communications adapter or a line of communications adapters, in the case of multi–line adapters.

The OSI Stack can share network access with other communication protocols. The default when using the ”minimum configuration” is to enable the use of all network accesses. For specific needs, it is possible to select only a subset of the network access.

Each network access is identified by means of a ”location code” which uniquely identifies a bus, a slot on the bus, a line on the adapter plugged into the bus, ... To list devices and location codes, use the command **lscfg**. The configuration is not automatically updated when an adapter is moved from one slot to another. See Adapter Location Codes, on page 2-2.

Within the OSI Stack, each network access can be identified by a logical name (called subdomain number). This name is used in the static RIB and transport relay to avoid to code the local address; the communications adapter can be changed without modifying the routing information. The relationship between network access and logical name is stored in the subnetwork objects. Default values are generated by the minimum configuration for subdomain number. They can be changed for convenience or, for example, to ”logically” swap communications adapters.

The relationship between network access and logical name is stored in a subnetwork object. This object contains the following information:

Adapter Code:	network access location code
Subdomain number:	logical name
State:	ON/OFF

The default state is ON. The state is significant only when using full CLNP profile and for ES–IS protocol. When the state is OFF it is not possible for these protocols to send or receive PDUs. If a RIB entry refers to a subnetwork in the OFF state, all the PDUs routed by this entry are discarded.

Network access is demonstrated in Figure 35.

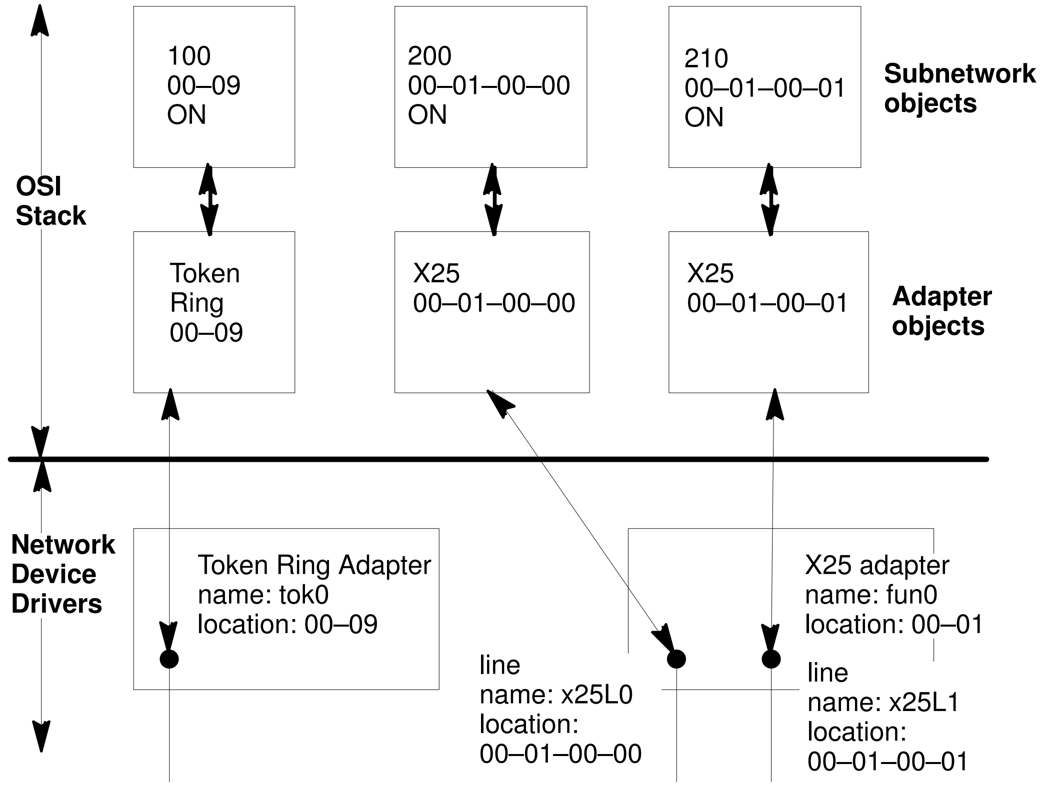


Figure 35. Network Access

Understanding PVC Configuration

When using PVCs, a configuration is required. The aim is two–fold:

- identify a PVC with a name (ASCII string), which will be used by an application to identify the X.25 line and channel number of the PVC.
- define who will receive data exchanged on a PVC. Possible recipients are COTP for all applications above transport layer (including transport relay) or X25.3 access method for applications above this interface (as for TPAD–HPAD).

Understanding Local NSAP Configuration

The OSI Stack can use any hexadecimal string with an even number of digits, in the range from 2 to 40 digits.

Its is recommended to use the following format which is derived from US GOSIP requirements:

Fixed prefix	Station identification	Selector (1 byte)

Selector (Sel) is a non–null value.

CAUTION:

1. There is a reserved prefix for NSAPs used by **NetShare**.
2. **NetShare** local NSAPs must not be defined as local NSAPs for the OSI Stack. See *NetShare User's Guide*, Bibliography on page B-1.

Understanding Dynamic Routing

Dynamic routing allows a station to automatically collect routing information and avoid the need for the administrator to statically define this data. It can only be used with full CLNP profile on LAN.

Limitations of dynamic routing and operation details are described in Use of LAN Subnetworks, on page 6-2.

In order to use dynamic protocol, it is mandatory top configure two parameters:

Use of the ES–IS protocol: must be YES

Type of function assumed on the system: ES (End System), IS (Intermediate System) or ES+IS

The IS function should be reserved for stations which interconnect LANs. On a single LAN, there is no need to define IS stations.

Understanding Static Routing Information Base Configuration

The static RIB contains routing information configured by the administrator. This information is used when operating with full CLNP or CONS profiles. It allows End Systems to know how to reach NSAPs located on remote stations, or Relay Systems to route incoming NPDU not destined for the local station. The relays are either the CLNP relay or the MSDSG relay, see Subnetwork Interconnections, on page 6-11.

To solve the routing issues, the static RIB contains a list of routes (also called RIB entries).

Each RIB entry contains the following information:

- Identification of a set of remote NSAPs
- Definition of remote SNPAs or relay NSAP
- Network to be used to access to the remote NSAPs (subdomain number)
- Miscellaneous.

The mechanism for address resolution by CLNS or CONS is shown in Figure 36.

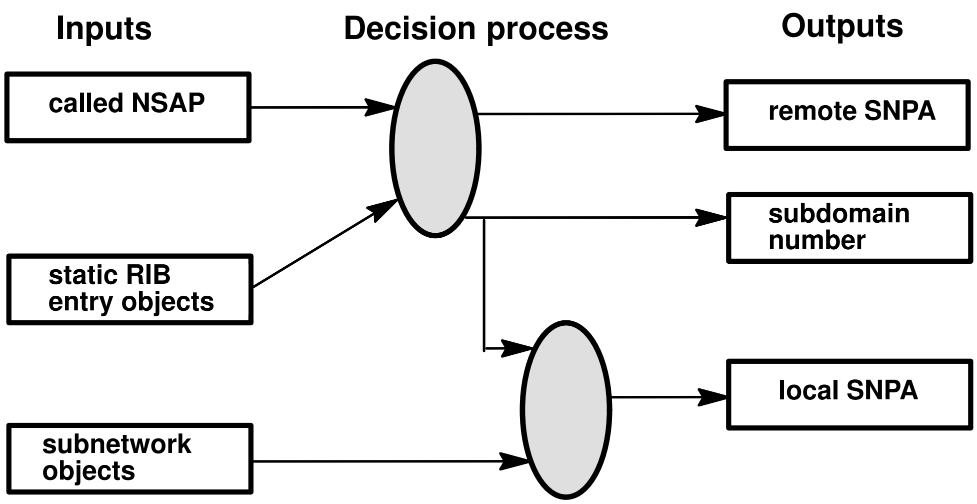


Figure 36. Address Resolution Mechanism by CLNS or CONS.

When the ES–IS protocol is active, the static RIB has a higher priority than the information gathered by ES–IS protocol.

Identification of a Set of Remote NSAPs (Remote NSAP name/type/mask)

The set of remote NSAPs is defined by three parameters:

- Remote NSAP,
- Remote NSAP Mask,
- Remote NSAP Type.

Remote NSAP

The called NSAP (i.e. the NSAP to be reached) is compared to the field remote NSAP to select the appropriate RIB entry.

Example

```
RIB entry
Remote NSAP name   : 0x12345678
```

The entry defines how to reach the NSAP 0x12345678.

Remote NSAP Mask

The remote NSAP mask is optional. It allows a set of NSAPs to be defined when a mask is defined. The remote NSAP Mask is a string with '0' and 'f' or 'F' characters. The length is either 0 (no mask) or the length of the Remote NSAP parameter. When a mask is defined, it must have the same length as the Remote NSAP parameter of the RIB entry. An **AND** operation is performed with the target NSAP and the mask. The result is compared with the Remote NSAP parameter in order to select the correct RIB entry. When defined, the remote NSAP must contain zeros for each zero of the mask, because of the comparison with the result of the **AND** operation.

When using masks, it is possible to have intersections between sets of remote NSAPs defined by several RIB entries.

Example: (Good practice – to be followed)

In the following example, all *target* addresses with a length of 8 digits and beginning with 0x1234 can be accessed via the routing path declared for this NSAP entry in the RIB. This type of description is mainly used for families of remote systems which are accessible through the same relaying systems.

```
RIB entries
Remote NSAP name  : 0x12340000
NSAP Mask : 0xFFFF0000
```

Both NSAPs, 0x12341234 and 0x1234 5678 match the same RIB entries.

Example: (Bad practice – to be avoided)

```
Entry 1:        remote NSAP     = 1230
                NSAP mask       = FFF0
Entry 2:        remote NSAP     = 1204
                NSAP mask       = FF0F
```

The NSAP 1234 matches both entries.

Remote NSAP Type

The Remote NSAP Type parameter is significant only for relaying CLNP and MSDSG. It defines which relay must be used. The following table shows the value to be used according to the incoming and outgoing profiles.

Incoming Profile	Outgoing Profile	NSAP Type
Full CLNP	Full CLNP	NSAP
Full CLNP	CONS	MSDSG
CONS	Full CLNP	NSAP or MSDSG (not supported for an X25 outgoing network)
CONS	CONS	NSAP or MSDSG

Remote SNPAs or Relay NSAP Definition (Network type/Connection type/Remote SNPA type/Remote SNPA)

Each RIB entry contains rules to compute a remote SNPA, that is used to access the system containing the target remote NSAP or a relay NSAP.

The parameters Network, Connection Type, Remote SNPA Type and Remote SNPA define the rules that must be used to compute the remote SNPA or the relay NSAP. The Remote SNPA parameter can contain six different types of information depending on the value of the other parameters. It can be a LAN address (12 hexadecimal digits), an X25 address (between 1 and 15 decimal digits), a PVC name, an NSAP name (hexadecimal string, even number of digits between 2 and 40), a LAN mask (with 12 successive F characters), or a WAN mask (with up to 15 successive F characters). The Remote SNPA parameters are shown in the following table:

Expected Result	Fields to Configure		
Remote SNPA	**Remote SNPA Type**	**Network**	**Connection Type**
LAN address (SNPA)	SNPA	ETH, TKRG, FDDI	NS
X25 address (SNPA)	SNPA	X25	SVC
PVC name	SNPA	X25	PVC
LAN mask	MASK	ETH, TKRG, FDDI	NS
X25 mask	MASK	X25	NS
relay NSAP	NET	NS	NS

NS: not significant

ETH=Ethernet, TKRG=Token Ring, FDDI=Fiber Distributed Data Interface

The syntax of the remote SNPA field is described below.

LAN Address (SNPA)

6 bytes, 12 hexadecimal digits.

See also osiadapterinfo (Adapter Information), on page 11-5.

X.25 Address (SNPA)

Between 1 and 15 decimal digits.

See also osiadapterinfo (Adapter Information), on page 11-5.

PVC Name

String of 1..8 characters.

Relay NSAP

NSAP syntax (hexadecimal character string with an even number of digits, between 2 and 40).

The relay NSAP is a pointer to another local RIB entry and does not necessarily exist on the local or remote systems. A single re–direction is possible. The referenced NSAP is a virtual NSAP which does not exist. It is only a way to refer to another local static RIB entry.

LAN Mask

String with 0 and F with the syntax 0*F*0*. The length of the string is equal to the length of the parameter Remote NSAP. The number of ’F’ characters is 12. The SNPA is the concatenation of the digits of the remote NSAP to be reached that match ’F’ characters in the mask.

Example:

```
Remote NSAP to be reached:    1234 56 78 9A BC DE F0
Mask:                         0000 FF FF FF FF FF FF
Remote SNPA computed:              56 78 9A BC DE F0
```

X.25 Mask

See LAN mask. For an X.25 address, the number of ’F’ characters is between 1 and 15.

Local SNPA (Subdomain)

The local SNPA to be used is identified by a subdomain number. The subdomain is a logical identification of the communications adapter. It has only a local meaning. To retrieve the calling SNPA (used to establish the connection) the system combines RIB information and the subnet definition.

Miscellaneous Parameters (NSAP priority/Static mode/State/Message lifetime/CUG)

Priority

This parameter defines a route priority. It is significant when masks are used and when there is intersection between sets of remote NSAPs defined by different RIB entries. In such cases, the route priority is defined by the:

- priority parameter
- mask selectivity (for RIB entries with the same priority parameter).

The mask selectivity is used to distinguish routes with the same priority parameter. A low value indicates a high priority, see Routing Framework and Routing Information Base, on page 3-10.

No mask or a mask which matches few NSAPs (i.e. with many FF) has a high priority.

RIB Part

The RIB part defines if the route is always available to protocol providers (STATICF), or if is stored in a cache (STATICV). For more information, see Routing Framework and Routing Information Base, on page 3-10.

It is recommended always to use the STATICF mode. STATICV mode entries are used only by CLNP protocol.

Because of static RIB implementation, see Routing Framework and Routing Information Base, on page 3-10, it is recommended to avoid intersections between fixed and variable entries, or between variable entries if they lead to the use of different routes. If such intersections exist, the selected route cannot be predicted.

State

Enables or disables a RIB entry.

MsgLifeTime

This parameter is reserved for CLNP. It allows the value of the message lifetime parameter, defined in CLNP (ISO 8473), to be defined for each route. This parameter is transmitted with each PDU and is decremented at least by one for each CLNP relay. When the value is null, the PDU is discarded. It is therefore recommended to configure a value which is 2 or 3 times the number of CLNP relays crossed.

CUG

This parameter is only significant for X.25. It specifies the CUG number accepted on the connection. This information is inserted in the *facilities* field of the call packets used to set up the virtual circuit, either with CLNP over X25 through SNDCF or with transport or CONS (i.e., using NSAP addressing).

CAUTION:

1. **Double entries are not permitted.** (The two RIB entries defined, as previously described, are considered as double if they have the same value for the Subdomain, Remote NSAP, Remote NSAP Mask and Remote SNPA parameters.)
2. **When several routes are defined for the same remote NSAP they are not used as ’back–up routes” in the case of network or connection establishment failure or when the subnetwork to be used is in the ”OFF” state.**

Static RIB Configuration Example

Figure 37 shows an example of static RIB Configuration:

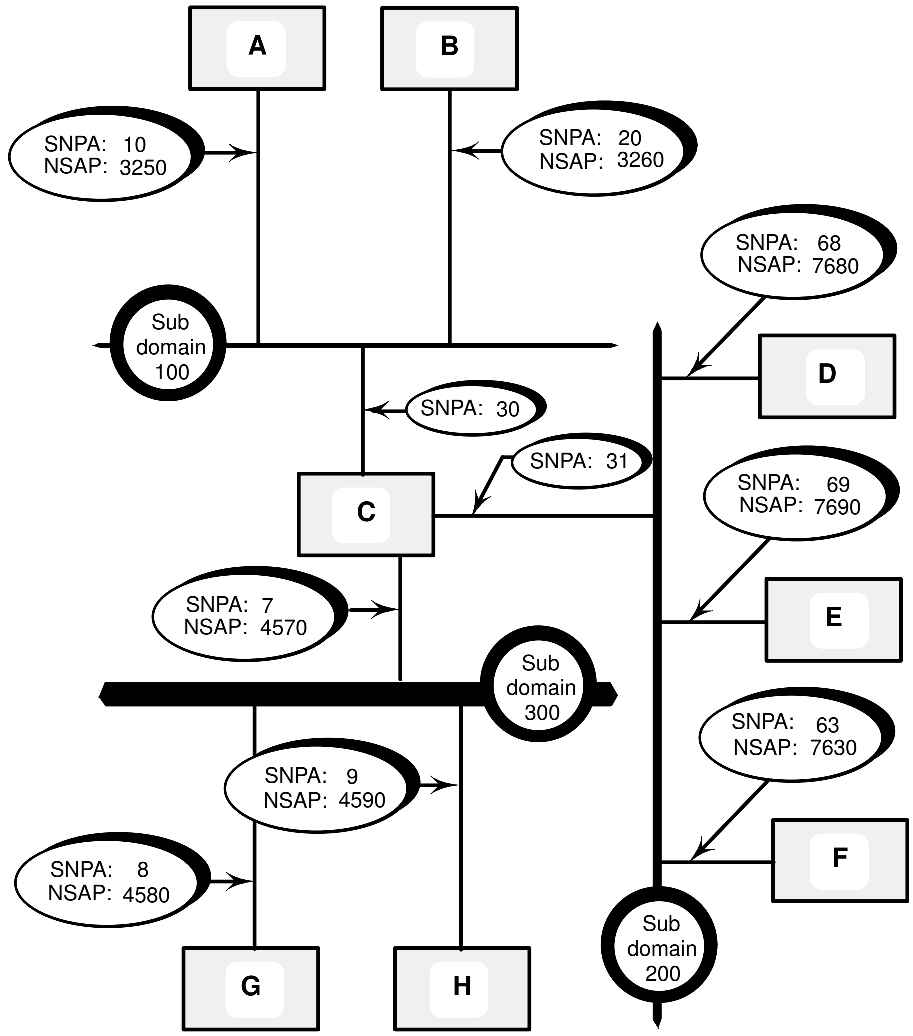


Figure 37. Static RIB Configuration

In this example, the administrator has chosen to attach the same subdomain number to the communications adapters using the same physical network. The SNPAs have strange values only for convenience of the example.

Note: The subdomain numbers are shown associated with the LAN segments for clarity. In reality these numbers have only a local significance.

The RIB entries for the example shown in Figure 37 are listed in the following table:

System	Remote NSAP Name	NSAP Mask	Subdomain Number	Remote SNPA
A	3260 4500 7600	 FF00 FF00	100 100 100	20 30 30
B	3250 4500 7600	 FF00 FF00	100 100 100	10 30 30
C	3250 3260 4500 7600	 FF00 FFF0	100 100 300 200	10 20 00F0 (mask) 0FF0 (mask)
D, E, F	7600 4500 3200	FF00 FF00 FF00	200 200 200	0FF0 (mask) 31 31
G, H	3200 7600 4500	FF00 FF00 FF00	300 300 300	7 7 00F0 (mask)

Subnetwork Table Entries

The Subnet Table entries for the example shown in Figure 37 are listed in the following table:

System	Subdo-main Number	Local SNPA
A	100	10
B	100	20
C	100 200 300	30 31 7
D	200	68
E	200	69
F	200	63
G	300	8
H	300	9

Understanding Transport Relay Configuration

The transport relay contains routing information configured by the administrator. This information is used to establish a new connection for each received incoming connection to be able to relay user data from one network to another.

To enable a connection to be established, the transport relay contains a list of **relay entry** records. Each record contains the following information:

- routing criteria
- routing information for the outgoing connection
- parameters to terminate the establishment of the incoming connection.

This mechanism is shown in Figure 38.

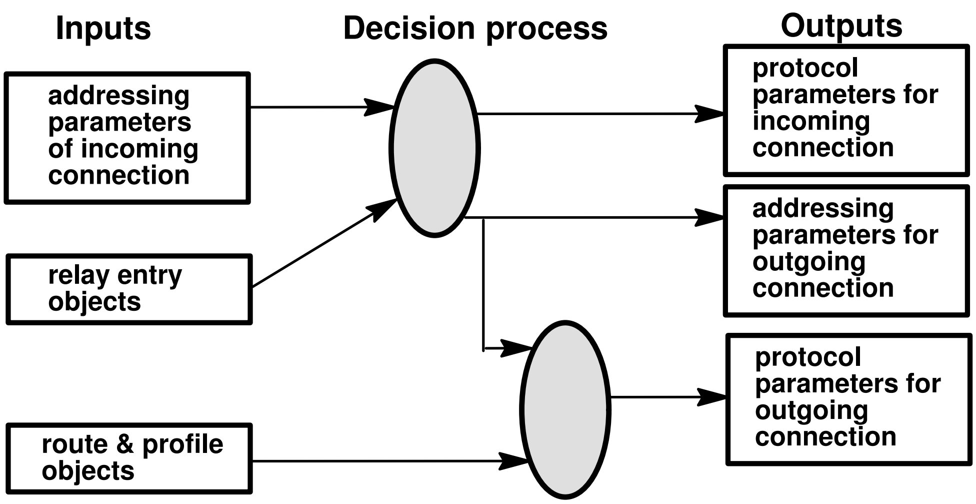


Figure 38. Transport Relay Configuration

Routing Criteria

The routing criteria is used to select a **relay entry** in the database. It is defined with the following information:

- `InTselEd` (incoming called TSEL)
- `InTselIng` (incoming calling TSEL)
- `InAddrIng` (incoming calling address)
- `InAddrType` (incoming address type)

'Incoming' refers to parameters of the incoming connection. The `InAddrIng` parameter is either an NSAP, a LAN MAC address, a X.25 DTE address or a PVC name, according to the `InAddrTyp` value (respectively NSAP, LAN, SVC, PVC). A convention for address fields ('**–**' minus character) means that the fields must be ignored in the comparison.

Example

```
InTselEd   :    1234
InTselIng  :    -
InAddrIng  :    -
InAddrType :    NSAP
```

Any connection to TSEL 1234 will match this relay route.

Parameter for Incoming Connection Negotiation

An additional parameter, called `InNegoClass` allows to respond to the incoming connection with a transport class which is lower than the proposed class.

The parameter is significant only for incoming X.25 connections. Its is recommended to preserve the default value of 4.

Routing Information for the Outgoing Connection

A relay route contains all the network and transport addresses necessary to establish the outgoing connection:

- `OutTselEd` (outgoing called TSEL)
- `OutTselIng` (outgoing calling TSEL)
- `OutAddrEd` (outgoing called TSEL)
- `OutAddrIng` (outgoing calling address)
- `OutAddrType` (outgoing address type)
- `LsapEd`

A convention using the '–' minus character allows the value of the incoming connection to be re–used.

The `OutAddrType` parameter is used to distinguish the address format and the profile to use:

CL: full CLNP (on LAN or X.25)

CLCO : CONS on X.25 or full CLNP on LAN

SVC : use X.25 SVCs and DTE address

PVC: use X.25 PVCs. In this case OutAddrEd is a PVC name and OutSubdIng is not significant

LAN: null CLNP profile with LSAP 0xFE

LANC: null CLNP profile. See Understanding LSAP Configuration, on page 8-30. When a LANC type is selected, it is possible to specify a value for the called LSAP. The calling LSAP is the configured default LSAP.

The `OutAddrEd` parameter is either an NSAP, a LAN MAC address, an X.25 DTE address or a PVC name.

If the outgoing connection uses NSAP addressing (CL or CLCO type) , the `OutAddrIng` parameters may contain the calling NSAP. For other than PVC profiles (SVC, LAN, LANC), the `OutAddrIng` parameter contains a subdomain name to identify the outgoing network access to use.

To establish the outgoing connection, the transport relay uses the outgoing called address to get the profile from the profile database.

See Understanding Profile and Route Configuration, on page 8-27.

Example

Figure 39 shows an example of the use of a transport relay.

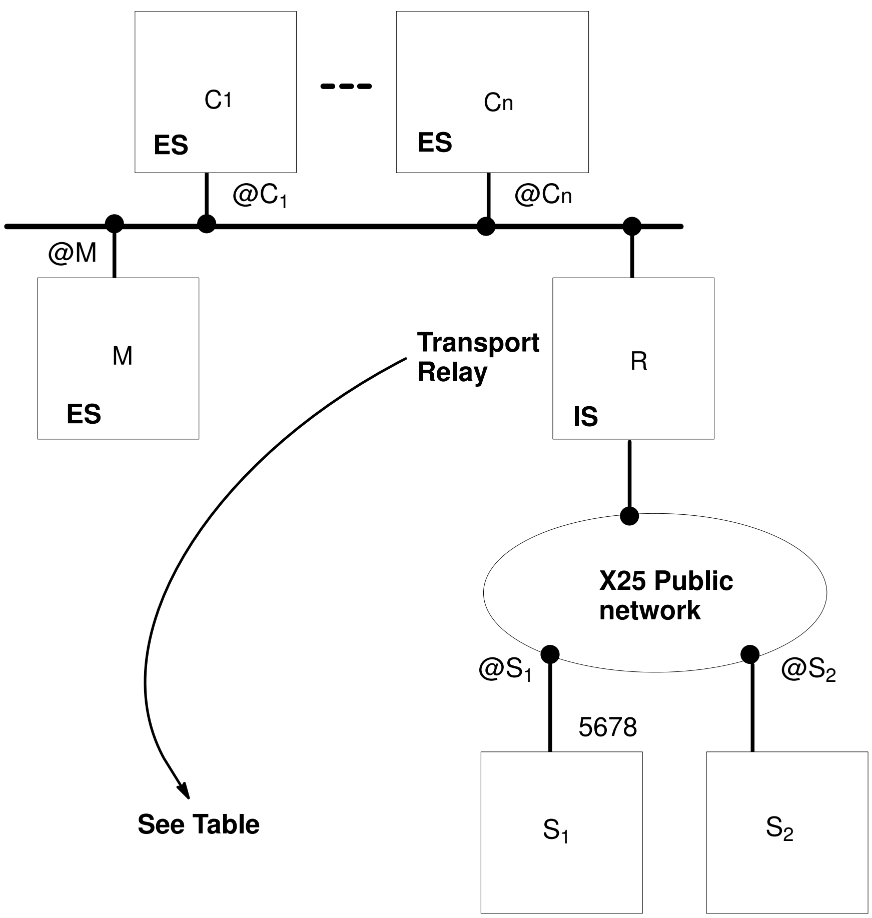


Figure 39. Use of Transport Relay

The relay configuration for system R is described in the table.

In Tsel Ing	In Tsel Ed	In Addr Ing	In Addr Typ	In Nego Class	Out Tsel Ed	Out Tsel Ing	Out Addr Typ	Out Addr Ed	Out Addr Ing
T_1	T	–	NSAP	4	–	–	SVC	100	@S_1
T_2	T	–	NSAP	4	–	–	SVC	100	@S_2
–	M	B	NSAP	4	–	–	SVC	100	@S_1

Client systems C_1 and master systems M are configured as ES and use the full CLNP profile. The relay is configured as an IS. The application on clients uses randomly local TSEL T_1 or T_2 and call TSEL T. The application on master calls TSEL M. Client workload must be shared between servers S_1 and S**2**, while the master accesses only S1. SVC profile (without NSAPs) is used over X.25.

Understanding Profile and Route Configuration

In order to establish a transport connection a lot of parameters are mandatory.

Transport parameters

These parameters are discussed in Protocol Parameters Configuration, on page 5-13. See also How to Add a Profile Entry From a Model, on page 9-80.

Transport class

Alternative Transport class

Use Expedited data

Use flow control

Use extended control

Use checksum

Priority

Initial credit

TPDU size

Use of alternate retransmission algorithm.

X.25 parameters

These parameters are discussed in Facility Management, on page 6-6.

CUG

CUD

For each transport connection request, the called address (NSAP, SNPA or PVC) is used to search for a profile or generate default values. The values generated are merged with the values defined by the application (or transport relay).

The profiles are identified by names. Two names have special meanings:

LAN profile

Refers to a default LAN profile defined by the administrator.

WAN profile

Refers to a default WAN profile defined by the administrator.

The relationship between addressing information, provided by the relay or by applications and the profile, is made using **route object**. A route object contains a remote address and a profile name.

The algorithm to retrieve connection parameters is as follows:

- search for a route

if a route is found

- merge known parameters and the parameters of the profile referenced by the route

else

- search for a default LAN or WAN profile

if a default profile is available

- merge the known parameters and the parameter of the profile selected

else

- generate missing parameters using default rules.

The default LAN or WAN profiles are retrieved using the address type:

PVC or SVC use the WAN default profile

any other format uses the LAN default profile.

Understanding Security Filter Configuration

There are two aims of the security filter configuration:

- define which data stream must be filtered (COTP transport connections, CLTP PDUs, relayed CLNP PDUs).
- define the list of authorized sources and destinations.

The filter defined by the administrator contains three items of data:

- called TSEL
- calling address (either NSAP, LAN address, X.25 DTE address, irrelevant for PVC)
- called address (either NSAP, LAN address, X.25 DTE address or PVC name).

A convention '–' minus character indicates that a parameter must be ignored.

The table explains how to configure filtering and which filter parameters are taken into account, depending on the data stream to be filtered.

	System type	**Local system is final system**		**Local system is a relay**	
	data stream type	connection oriented transport protocol	connectionless transport protocol	CLNP relay	MSDSG or transport relay
Configured filter	COTP	ON	OFF	OFF	ON
	CLTP	OFF	ON	OFF	OFF
	CLNP	OFF	OFF	ON	OFF
Parameter	called TSEL	X	X	Not significant	X
	calling address	X	X	X	X
	called address	X	X	X	X

Understanding LSAP Configuration

The LSAP is used on LAN to select a communications stack (a protocol) on top of the LLC layer. LSAPs are expressed in even bytes.

The default value for OSI protocols is 0xFE. This value is used for full CLNP profiles.

The OSI Stack can be configured to receive a list of LSAPs which are named ”null CLNP LSAPs”. It is assumed that such PDUs correspond to the null CLNP profile.

By default, this list contains only LSAP 0x20, to support OSI/DSA profiles.

The first LSAP of the list is known as the **default configured LSAP**. It is used as both calling and called LSAP when the profile does not use the LSAP 0xFE.

Understanding Netbios Configuration

Netbios is a protocol which is usually offered on top of TCP/IP, IPX/SPX or on NetBui. Netbios for AIX supports these three Transport protocols. Nevertheless, specifications have been defined by International bodies to run Netbios on top of OSI Transport, using specific communications profiles.

The aim of the Netbios configuration for OSI is to enable the use of these profiles, even though Netbios for AIX does not support OSI Transport.

These profiles are therefore reserved for advanced users.

Note: You are recommended to leave the initial configuration unchanged and not to use the NSAP and LSAP which are configured for Netbios.

Understanding Stack Tuning

There are three main ways to tune OSI communications:

- modify global parameters, see Table of Parameter Descriptions, on page 8-5
- modify parameters independently on each connection
- modify communications adapter or system configuration.

The tuning on a connection basis requires the application configuration or profile definition to be modified. The tunable parameters are explained in Protocol Parameters Configuration, on page 5-13. See also Understanding Profile and Route Configuration, on page 8-27.

Refer to Operating System and Driver Configuration, on page 8-34 for information on how to tune communications adapters and the operating system.

The global parameters are part of the OSI Stack configuration and are described below.

Basic Configurable Parameters

Maximum Number of Connections (COTP, COSP, COPP)

This parameter enables the maximum number of simultaneous connections, maximum amount of memory consumed and the system workload to be limited.

Transport Connections Multiplexing Rate

When operating over X.25 with classes 2, 3 or 4, the transport may multiplex several connections on a single X.25 virtual circuit (providing that they use the same calling / called X.25 addresses, same facilities, CUD, connection priority and that the alternate class is not 0).

If there is an X.25 attachment with few virtual circuits, it is recommended to select a high multiplexing rate.

The impact on performances depends on the facilities negotiated for the virtual circuit. For example, when a low throughput is negotiated, transport throughput will be higher using a low multiplexing rate. If the X.25 throughput is not the limiting parameter, multiplexing is recommended.

Advanced Parameters for Tuning

You are recommended to keep the default values in order to maintain optimum performance.

ES–IS Timers

These timers define the frequency of ESH or ISH PDUs emission. See Use of LAN Subnetworks, on page 6-2.

Reducing the delay permits the detection of network re–configurations (as for new stations, for example). Too small a value may overload the network and reduce the available throughput.

Session Connection / Disconnection Timers

These timers define the maximum delay allowed to establish or release a connection. A small value will reduce the delay of the connection phase; in the event of a problem, the connection will be released quickly.

The parameters do not modify data transfer performances.

Session Indication Timer

Session connection indications are stored by the session provider waiting for a request of a server to handle the connection. This parameter defines the maximum delay during which a indication is stored. After this delay the connection is refused.

For overloaded servers, a high value may reduce the number of connections refused because the server is not ready.

Note: Stored connections consume resources.

Transport T1 Timers

The timers define the delay between the re–transmission of un–acknowledged PDUs. This delay is relevant only for connections using class 4.

A small value reduces the delay to detect a network failure and improves performance. It also increases network load and the risk of increasing the number of erroneous detections of problems.

LAN and WAN networks use different values to enable the finest tuning.

Maximum Number of Transport Transmissions

This parameter defines the maximum number of retries before indicating a network failure. A small value reduces the delay to detect a failure, but may increase the risk of erroneous diagnostic. It does not impact performance.

Buffer Memory Size

This parameter defines the maximum amount of memory used by the OSI Layers. A first limit is defined in order to prevent these protocols consuming all of the system memory; when this limit is reached, all the OSI communications are stopped.

The default value is usually enough for normal use. A too–small value may reduce performance. It may also lead to closing the connection. In the extreme case, a fatal lack of memory can occur. In this event, the OSI Stack and all OSI applications must be stopped and restarted.

Operating System & Driver Configuration

This section describes certain system parameters which are recommended to be modified before using the OSI Stack.

Memory Management

The memory used by the OSI Stack cannot be predicted: it depends on the system and communication workload as well as network traffic.

It is possible to define a limit for the memory allocated by the OSI Lower Layers to prevent the OSI Stack consuming all of the System memory. When using ”automatic” mode, the limit is widely over–estimated for normal use.

The formula used to calculate the limit integrates the number of transport connections and X.25 line configurations. It is automatically updated when one of these parameters is modified.

The limit calculated can be controlled using the **osisnapshot** command, see page 10-33. This command also displays the memory used, allowing the administrator to adapt the system and stack configuration. The administrator may adapt the size of the paging space and define an explicit limit for OSI Lower Layers.

Paging Space Configuration

The default configuration of paging space should cover most of the OSI Stack requirements. Nevertheless, it is recommended to check the swap paging space used during ”normal operational state” or during peak phase and to adapt the paging space size, using the command **lsps –s**.

When the OSI Stack is running , if no more memory or paging space is available, the stack will crash and the following message will appear:

```
Add paging space... telecom server out of order
```

The following is also written in the MAD trace, see OSI Diagnostic Interactive Toolkit (ODIT) User’s Guide, listed in the Bibliography on page B-1.

```
-----------------------------------------------------------------
 <time stamp>      |  MAD
-----------------------------------------------------------------
        > mem_alloc : refused allocation
        > needed <P1> pages , only <P2> available
        > need still <P3> bytes of paging space
        > ADD PAGING SPACE !!!!
```

where

<time stamp> is the time parameter.

Parameter <P1> is the total number of pages required,

<P2> is the number of free pages,

<P3> is the memory size required.

The system administrator must:

- increase the paging space size by at least <P3> bytes, see How to Show Memory Buffer Status, on page 9-124,
- unload the Stack, see Stopping the OSI Stack, see page 8-37,
- reload the Stack, see Starting the OSI Stack, see page 8-36.

LAN Device Driver Configuration

If performance problems are detected, it is recommended to increase the transmit queue size of the LAN device driver used.

Example: lsattr –E –l ent0

Starting & Restarting the OSI Stack

Starting the OSI Stack

Once you have configured the stack, you can activate it through SMIT (**smit ConfOSILoad**, see page 9-113) or with a command **osiload**, see page 10-15.

The OSI drivers and the stack framework will be loaded in the kernel, the stack will be configured with the information stored in the ODM data base.

This phase will also active all the applications register by the Stack Applications Support. See **osisas Command**, on page 11-7. Refer to each application manual, see Bibliography, on page B-1, to know if the application is registered, or if you have to start it manually.

Automatic Restart of OSI Stack

It is usual for the Stack is be automatically restarted during the boot phase.

To disable the automatic restart, the OSI Stack must be stopped before rebooting the system. See Stopping the OSI Stack, on page 8-37.

To permanently remove automatic stack activation, edit the */etc/inittab* file and set the **osidaemon** entry to **off**. See After Installing the OSI Stack, on page 7-7.

Stopping the OSI Stack

Before stopping the OSI Stack, you must stop all the applications which are not registered by the Stack Applications Support (SAS), see page 8-38. Registered applications are automatically stopped.

When all applications are stopped, you can stop the stack using SMIT (**smit ConfOSIUnload**, see page 9-112), or with the **osiunload** command, see page 10-35.

Using Stack Applications Support

Purpose

The Stack Applications Support (SAS) allows applications to be started using OSI Services after stack activation. These same applications are stopped (in reverse order to starting) before the stack is stopped using the **osiunload** command, on page 10-35.

Applications are not pre–defined and must be registered by the SAS.

Most of the Bull applications use this support and register themselves during the installation and configuration phase.

This section explains how to register other applications, developed by the customer.

The administrator can list the registered applications and enable or disable each one using **smit OSISAS,** see Accessing OSI Start Applications Support on page 9-33 or using the **osisas command** on page 11-7.

ODM Object Syntax

```
class ISO_APPLICATIONS
{

        char Name [20];
        char Ident [20];
        method Start [256];
        method Notify [256];
        method Stop [256];
        method TestStop [256];
        long SortOrder;
        long Disable;
        long NoMessage;
};
```

Description

The osi_frame package contains the definition of the specific ODM object ISO_APPLICATIONS used by the *osilinkd* daemon, or by the **osisas command**.

In order to run an application with the SAS mechanism, an object of this class must be added (normally during the installation phase of the package). The class is stored in the OSI database directory (/etc/objrepos/osi).

Fields

Name
Name of the object, displayed when start and stop methods are used.

Names must be unique with the current SMIT database.

Value is a string with a maximum length of 20 characters.

Identifier
Links several objects to one application. Must be common to all objects of an application.

Value is a string with a maximum length of 20 characters.

Start
Command to start the application.

Value is a string with a maximum length of 256 characters.

Notify

Command to notify the application that the stack is going to stop.

Value is a string with a maximum length of 256 characters.

Stop

Command to stop the application.

Value is a string with a maximum length of 256 characters.

TestStop

Command which returns 0 when the application is completely stopped, else 1.

Value is a string with a maximum length of 256 characters.

Note: This field should be used to bypass problems resulting from a random time elapse needed to stop an application, see Example of SAS Management with FTAM, on page 8-40.

*** SortOrder**

Used to start different object methods in increasing order and stop them in decreasing order.

Value is an integer.

Values in the range from 0 to 9 are reserved for the OSI Stack.

Values in the range from 10 to 99 are reserved for Bull Applications.

Mandatory.

Disable

Corresponds to the SAS State (ON or OFF).

0 = object taken into account.

Not 0 = object ignored.

Value is an integer.

NoMessage

Corresponds to the message display state (ON or OFF).

0 = message displayed when starting and stopping application.

Not 0 = no message displayed.

Value is an integer.

Example of SAS Management with FTAM

An ISO_APPLICATIONS object is defined for SAS management of FTAM. This object is added during the FTAM delivery installation (and removed during its de–installation).

Object

File *ld_appli_ftam.add*

```
ISO_APPLICATIONS:
        Name = "ftam"
        Identifier = "ftam"
        Start = "/usr/bin/ld_ftam start >/dev/null 2>&1"
        Notify = ""
        Stop = "/usr/bin/ld_ftam stop >/dev/null 2>&1"
        TestStop = "/usr/lpp/ftamx/ftam_running"
        SortOrder = 10
        Disable = 0
        NoMessage = 0
```

Packaging file

In the installation script file, we find:

```
.../...
ODMDIR=/etc/objrepos/osi
odmdelete -o 'ISO_APPLICATIONS' -q 'Identifier=ftam' >/dev/null 
2>&1
odmadd /usr/lpp/ftamx/ld_appli_ftam.add > /dev/null 2>&1


.../...
```

In the de–installation script file, we find:

```
.../...
ODMDIR=/etc/objrepos/osi
odmdelete -o 'ISO_APPLICATIONS' -q 'Identifier=ftam' >/dev/null 
2>&1
.../...
```

Now Let's Go

You are now familiar with OSI concepts and OSI Stack features.

Here is a check list of what you have to do:

1. Define network topology and locate applications on systems.
2. Identify stations which will have a relay function.
3. Configure the communications adapter for each station especially X.25 adapters.
4. Use the table, see OSI Applications – OSI Layer Requirements on page 5-3 to install the appropriate software packages (see also Installing the OSI Stack on page 7-1) and install licenses.
5. If NSAP addressing is preferred, define a structured addressing plan. Collect addressing information for all stations using the "Station Worksheet" on page 8-43.
6. Identify all communications need between applications.
7. Complete "Calling Application Worksheet" on page 8-45.
8. Complete "Relay Worksheet"on page 8-47 for each relay between the pair of applications.
9. Complete "Called Application Worksheet" on page 8-51.
10. Configure the stations which have calling or called applications and use the information registered in the "Station Worksheet", "Calling Application Worksheet" and "Called Application Worksheet". Follow chart "Stack Addressing Configuration" or actions specified on worksheet.
11. Configure relays using information registered in "Station Worksheet" and "Relay Worksheet". Use chart "Relay Configuration" or follow actions specified in worksheet.
12. That's all – your system is ready to go.

Additional information concerning installation is to be found in:

- OSI Stack Installation, on page 7-1.
- the Software Release Bulletin (SRB) file.

Additional information concerning configuration is to be found in Configuring, Starting & Stopping the OSI Stack, on page 8-1.

If you have problems using the stack configurator, refer to Configuring the OSI Stack Using SMIT, on page 9-1 and Configuring the OSI Stack Using Commands, on page 10-1.

If you cannot communicate, read the *OSI Diagnostic Interactive Toolkit (ODIT) User's Guide*, see Bibliography, on page B-1.

Worksheets for OSI Sites

The following worksheets are provided to help you prepare your OSI sites. SMIT FastPaths provide the links to the appropriate configuration menus.

Fill in the boxes which concern your installation. The boxes provided do not necessarily determine the number of entries.

- Station Worksheet, on page 8-43.
- Calling Application Worksheet, on page 8-45.
- Relay Worksheet, on page 8-47.
- Called Application Worksheet, on page 8-51.

Station Worksheet

Station Worksheet 1/2

Local Addressing

Station Name

Local NSAPS

Configure local NSAP
smit ConfOSIAdrLoc

Local NSAP Name	TYPE Local / Group

Network Access

Configure comms. adapters used
smit ConfOSIAdapt

Adapter Code	Address	Subdomain number	MAXNPDU

PVCs

Configure PVC
smit ConfOSIPVC

Adapter code	PVC Name	LCN	Managed by (module ownner)
			COTP / MAX3
			COTP / MAX3
			COTP / MAX3

Limits

Configure limits

smit ConfOSILayers

Max. no. of connections | | Default = 64

Dynamic Routing

Configure ES–IS protocol

smit ConfOSILayers

Use of ES–IS protocol? YES / NO

If YES

Type of function assumed on the system ES / IS / ES+IS

Security Control

Data Stream to Filter

Configure security controls

smit ConfOSISecurity

COTP connection establishment ON / OFF

CLTP received PDUs ON / OFF

CLNP Routed PDUs ON / OFF

Authorized Addresses

Configure security filters

smit ConfOSIFilters

Filter Name	Called TSEL (1)	Address Type (2)	Called Address	Calling Address

(1) **not significant for CLNP PDUs**

(2) **one of NSAP / LAN / SVC / PVC**

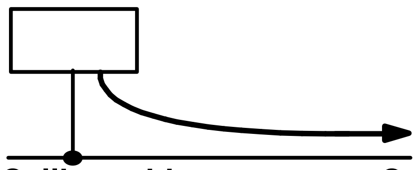
Calling Application Worksheet

Calling Application Worksheet 1/2

Worksheet reference	____________________
Name of Local Station	____________________
Name of Final Station	____________________
Communications profile used by the local application	LAN / LANC / SVC / PVC / CL / CLCO

LAN, LANC or SVC Profile

configure application



Calling address	**Called address**
____________________	____________________

PVC Profile

configure application

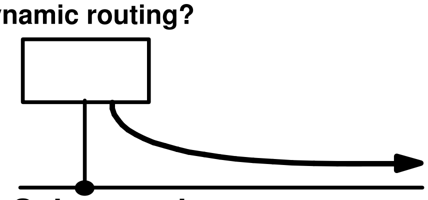
PVC name	____________________

CL or CLCO Profile

configure application

Local NSAP	____________________
Called NSAP	____________________
Use of dynamic routing?	YES / NO

If NO



Subnetwork	**Called NSAP**	**Called address**
____________	____________	____________

configure static RIB
smit ConfOSIAdrRIB

Remote NSAP name	**Called NSAP**
NSAP type	**NSAP**
Network type	**X25 / FDDI / Ethernet / Token Ring**
Connection type	**SVC**
Remote SNPA type	**SNPA**
Remote SNPA	**Called address**
Subdomain number	**Subnetwork**

Transport Addresses (Optional)

configure application

Calling TSEL ____________________

Called TSEL ____________________

Transport Parameters

configure application

	Options	Defaults SVC PVC	Defaults LAN LANC CL CLCO
Transport class	0 / 2 / 3 / 4	2	4
Alternate transport class	0 / 2 / 3 / 4	0	not applicable
Use expedited data?	YES / NO	YES	YES
Use flow control?	YES / NO	YES	YES
Use extended format?	YES / NO	NO	NO
Use checksum?	YES / NO	NO	NO
Priority	MAX / HIGH / MEDIUM / LOW	LOW	LOW
Credit 1..15	____________	15	15
TPDU size	____________ 128, 256, 512, 1024, 2048, 4096, 9192	8192	8192

X.25 Parameters

CL, CLCO Profile

configure static RIB

smit ConfOSIAdrRIB

CUG ____________________

SVC Profile

configure application

Facility name	**Value**
____________	____________
____________	____________
____________	____________

Next Station

Final station? YES / NO

If YES

Next worksheet is **Called Application Worksheet**

Else

Next worksheet is **Relay Worksheet**

Reference of next worksheet ____________

Relay Worksheet

Relay Worksheet 1/4

Worksheet reference ____________________

Name of Local Station ____________________

Name of Next Station ____________________

Addresses and Profiles

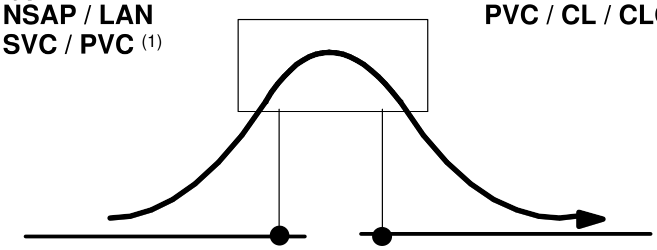
Local information

incoming address type
NSAP / LAN
SVC / PVC (1)

outgoing profile
LAN / LANC / SVC /
PVC / CL / CLCO

Calling application

Called application



calling address ________________

incoming subnetwork ________________

outgoing subnetwork ________________

called address ________________

calling NSAP ________________

outgoing PVC name ________________

called NSAP ________________

outgoing calling NSAP ________________

calling TSEL ________________

incoming called TSEL ________________

called TSEL ________________

outgoing calling TSEL ________________

(1) **NSAP corresponds to CL or CLCO profile, LAN corresponds to LAN or LANC profile**

CLNP or MSDSG Relay

Relaying from Calling to Called Application

Use of dynamic routing? YES / NO

If NO

Configure static RIB
smit ConfOSIAdrRIB

Remote NSAP name	**Called NSAP**
NSAP type	**NSAP / MSDSG**
Network type	**X25 / FDDI / Ethernet / Token Ring**
Connection type	**SVC**
Remote SNPA	**Called address**
Subdomain number	**Outgoing subnetwork**

Relaying from Called to Calling Application

Use of dynamic routing? YES / NO

If NO

Configure static RIB
smit ConfOSIAdrRIB

Remote NSAP name	**Calling application**
NSAP type	**NSAP / MSDSG**
Network type	**X25 / FDDI / Ethernet / Token Ring**
Connection type	**SVC**
Remote SNPA	**Calling address**
Subdomain number	**Incoming subnetwork**

Transport Relay

Configure transport relay
smit ConfOSIAdrRelays

InTselEd	**Incoming called TSEL**
InTselIng	**Calling TSEL**
InAddrType	**Incoming address type**
InAddrIng	**Calling NSAP / calling address**
InNegoClass	**4**
OutTselEd	**Called TSEL**
OutTselIng	**Outgoing calling TSEL**
OutAddrType	**Outgoing profile**
OutAddrEd	**Called NSAP / called address / outgoing PVC**
OutAddrIng	**Outgoing calling NSAP / outgoing subnetwork**

Outgoing Transport Relay Connection

Configure addressing for outgoing connection

If CL or CLCO profile

Use of dynamic routing? YES / NO

If NO

Configure static RIB
smit ConfOSIAdrRIB

Remote NSAP	**Called NSAP**
NSAP type	**NSAP**
Network type	**X25 / FDDI / Ethernet / Token Ring**
Connection type	**SVC**
Remote SNPA type	**SNPA**
Remote SNPA	**Called address**
Subdomain number	**Outgoing subnetwork**

Configure protocol parameters for outgoing connection

		Defaults: SVC PVC	Defaults: LAN LANC CL CLCO
Name of the profile	________		
Transport class	**0 / 2 / 3 / 4**	2	4
Alternate transport class	**0 / 2 / 3 / 4**	0	not applicable
Use expedited data?	**YES** / **NO**	YES	YES
Use flow control?	**YES** / **NO**	YES	YES
Use extended format?	**YES** / **NO**	NO	NO
Use checksum?	**YES** / **NO**	NO	NO
Use alternate retrans. algorithm?	**NO**	NO	NO
Priority	MAX / HIGH / MEDIUM / LOW	LOW	LOW
Credit (If X25 outgoing network)	1..15 ________	15	15
CUD	________		
CUG	________		

If non–default values are used, create a profile
smit ConfOSIAdrProfiles

AND
Create a route
smit ConfOSIAdrRoutes

Profile name	**Name of the profile**
Type of the route	**NSAP / LAN / SVC / PVC**
Route Value	**Called NSAP / called address / outgoing PVC name**

Addressing for Incoming Transport Relay Connection using NSAP Address

Use of dynamic routing? YES / NO

If NO

Configure static RIB
smit ConfOSIAdrRIB

Remote NSAP	**Calling NSAP**
NSAP type	**NSAP**
Network type	**X25 / FDDI / Ethernet / Token Ring**
Connection type	**SVC**
Remote SNPA type	**SNPA**
Remote SNPA	**Calling address**
Subdomain number	**Incoming subnetwork**

Next Station

Final station? YES / NO

If YES

Next worksheet is **Called Application Worksheet**

Else

Next worksheet is **Relay Worksheet**

Reference of next worksheet ______________

Called Application Worksheet

Called Application Worksheet

Worksheet reference ______________________

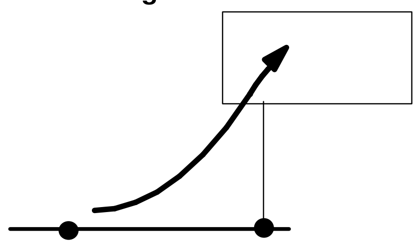
Name of Local Station ______________________

Communication Profile **LAN / LANC / SVC / PVC / CL / CLCO**

CL, CLCO Profile

Use of dynamic routing? YES / NO

If NO



Calling address ______________ **Subnetwork** ______________

Calling NSAP ______________

Configure static RIB

smit ConfOSIAdrRIB

Remote NSAP	**Calling NSAP**
NSAP type	**NSAP**
Network type	**X25 / FDDI / Ethernet / Token Ring**
Connection type	**SVC**
Remote SNPA type	**SNPA**
Remote SNPA	**Calling address**
Subdomain number	**Subnetwork**

Chapter 9. Configuring the OSI Stack Using SMIT

Configuring the OSI Stack Using SMIT Overview

This section provides information about the OSI Configurator. You can find more information in:

- National Language Support on page 9-1.
- Help on page 9-1.
- OSI Stack Configurator Menu structure on page 9-2.
- Parameter Management – Quick reference on page 9-26.

National Language Support

The OSI Stack configurator includes support of the national language. A catalog is provided for the US English language (En_US).

Help

On–line **Help** is available for all fields. The environment variable LANG must be set to 'en_US' to use the US English catalogs.

OSI Stack Configurator Menu Structure

This section describes the configurator menus in graphic form, showing inter–dependencies, **FastPath** access, section and page references. Management functions sorted in alphabetical order are also given as a method of quickly finding the information you want.

You can find more information in:

Menu Structure –Configuration Management

Before attempting to manage the OSI Stack using the SMIT configurator, it is important to understand the menu structure.

There are five levels of configuration management:

1. Manipulation, as a block: creation, selection, removal and display.
2. Definition: with configurable parameters within a configuration.
3. Activation, de–activation of kernel drivers.
4. Memory actions: display, reset RIBs.
5. Minimum configuration, a basic facility including levels 1. and 2.

In addition, two other facilities are available:

1. OSI Start Applications Support for Application start/stop management.
2. OSI Diagnostic Interactive Tool for problem determination, see *OSI Diagnostic Interactive Toolkit (ODIT) User's Guide*, Bibliography, on page B-1.

The hierarchical structure of the configurator menus is shown in Figure 40.

Note: Menus or attributes in a dialog can be hidden if some packages are not installed.

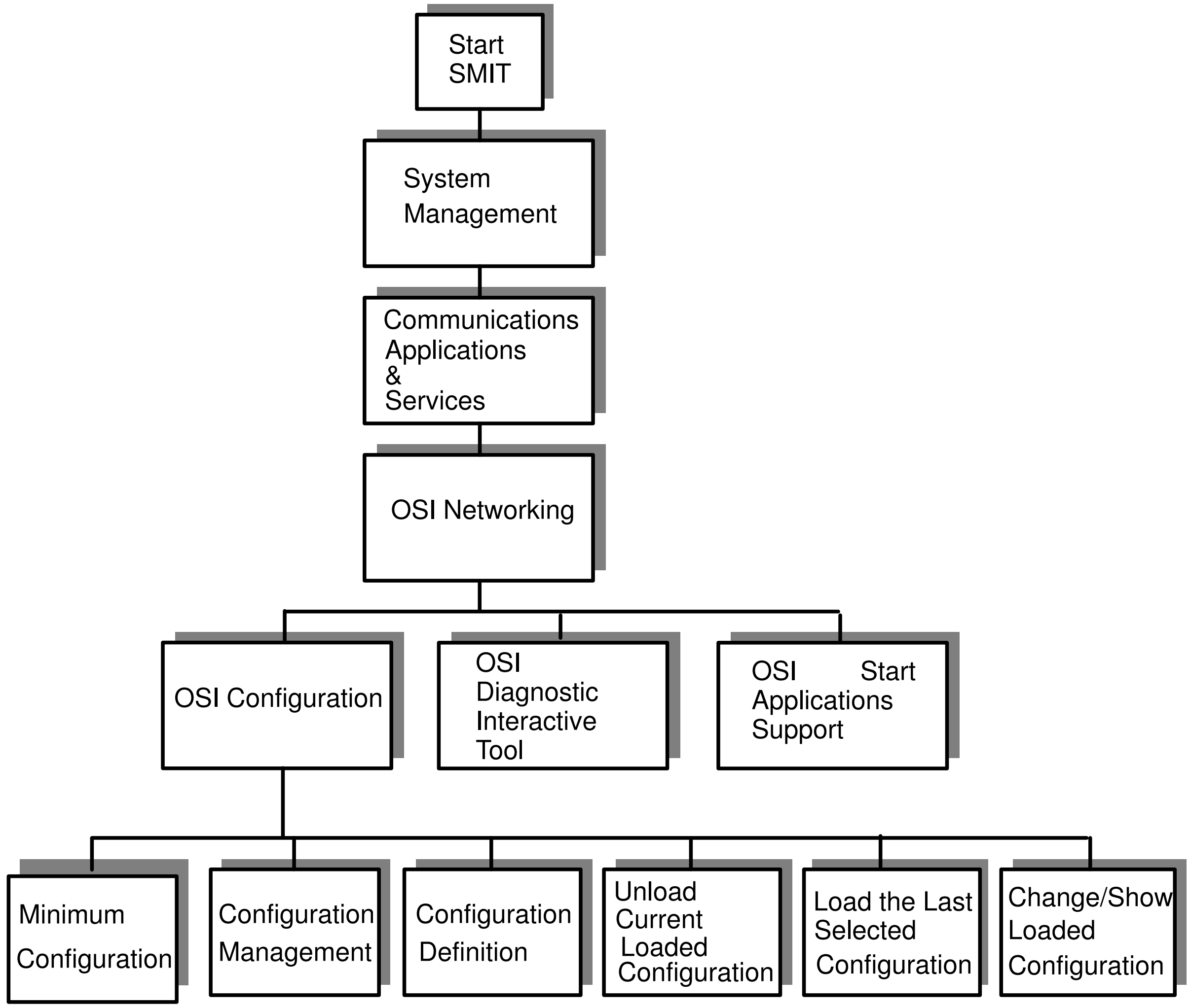


Figure 40. Configurator – Menu Flow.

Startup, System Management, OSI Networking & Configuration Menus

The Startup and System Management menus allow you to start up SMIT in the environment of your choice, and then gain access to the OSI Configuration menu via the System Management and OSI Networking menus.

Figure 41 shows the Startup, System Management, OSI Networking & Configuration menus:

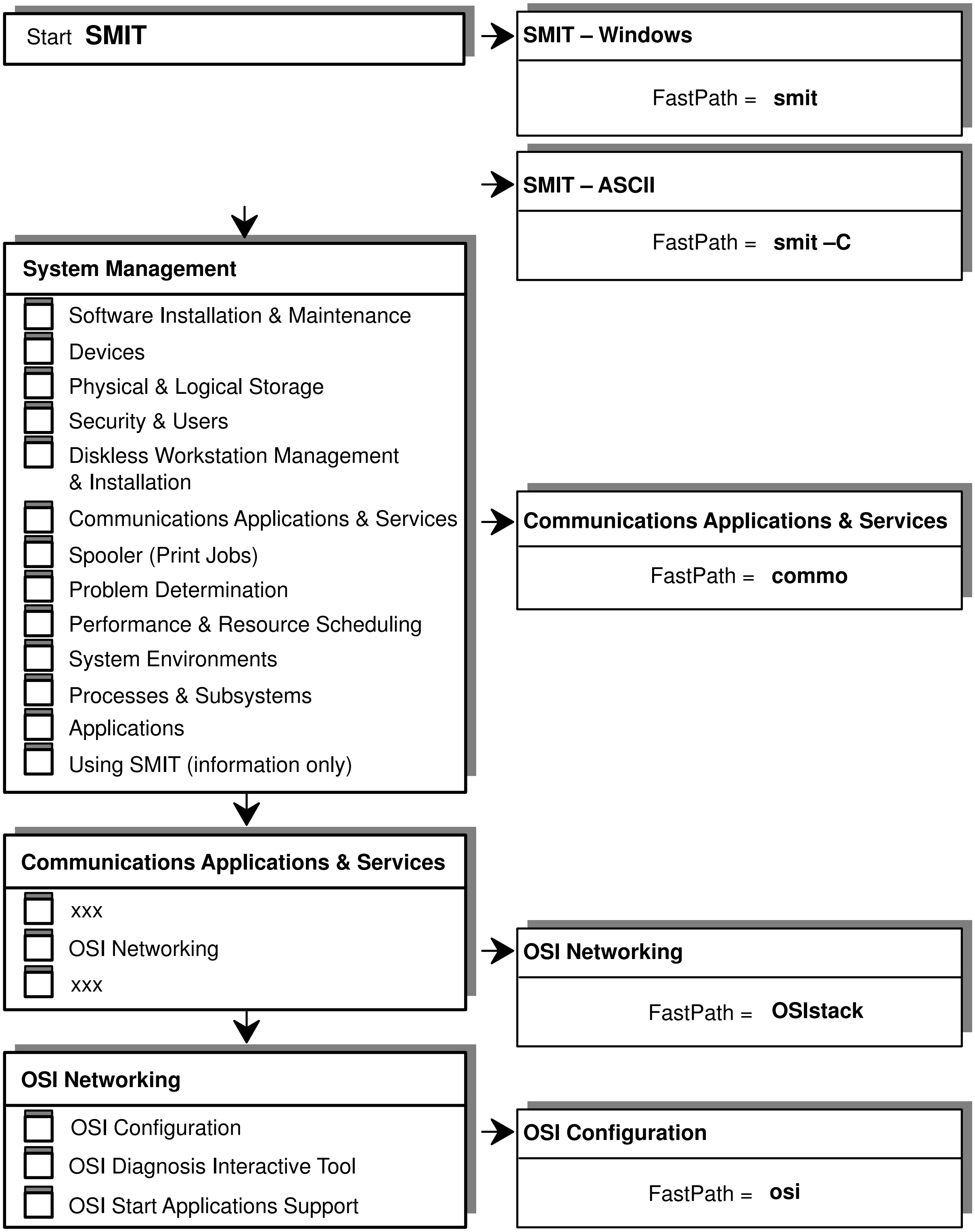


Figure 41. Configurator – Startup, System Management, OSI Networking & Configuration Menus.

OSI Configuration Menu

The starting–point for Stack management within the menu structure is the OSI Configuration menu. This menu displays all the options used for management of the configurations.

Figure 42 shows the OSI Configuration menu:

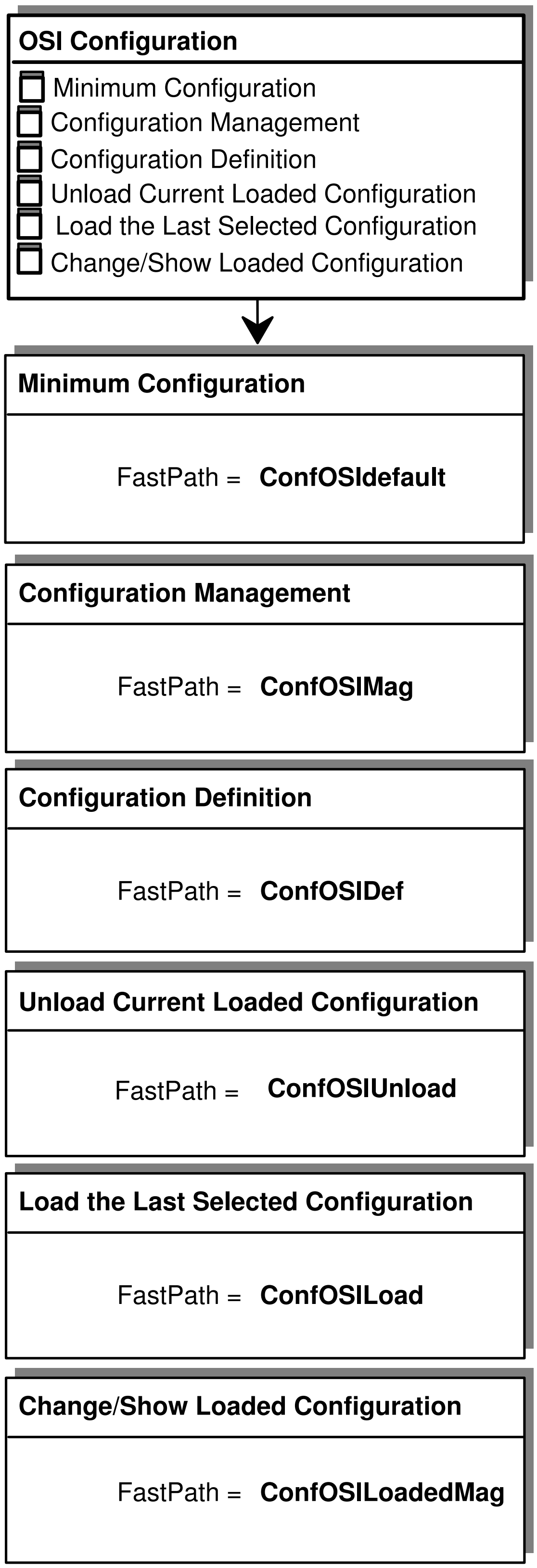


Figure 42. Configurator – OSI Configuration Menu.

Minimum Configuration Menu

From the OSI Configuration menu, this menu provides a default configuration defined by the system administrator.

Figure 43 shows the Minimum Configuration menu:

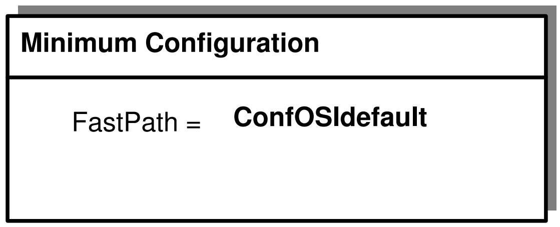


Figure 43. Configurator – Minimum Configuration Menu.

Configuration Management Menu

From the OSI Configuration menu, this menu permits management of all OSI configurations. All users may list, display or select any of the system configurations whether they are active or not. Additionally, the system administrator (a super–user) may create, change or delete configuration names or comments.

Configurations are identified by a unique name. The creation of a new configuration requires the definition of a new name. This new configuration may contain only default values, or may be a copy of another configuration.

Figure 44 shows the Configuration Management menu:

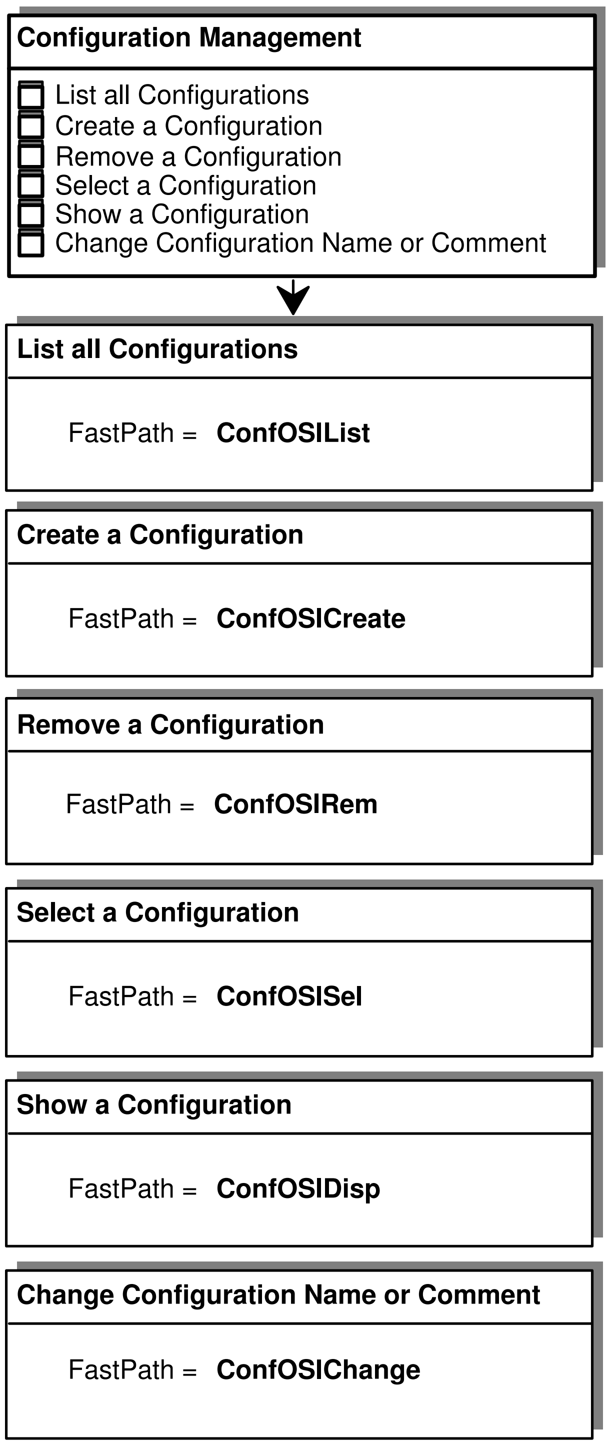


Figure 44. Configurator – Configuration Management Menu.

Configuration Definition Menu

From the OSI Configuration menu, the Configuration Definition menu opens the path to a series of menus permitting the selected configuration to be defined (list, add, change remove – one or several parameters within the selected configuration).

Figure 45 shows the Configuration Definition menu:

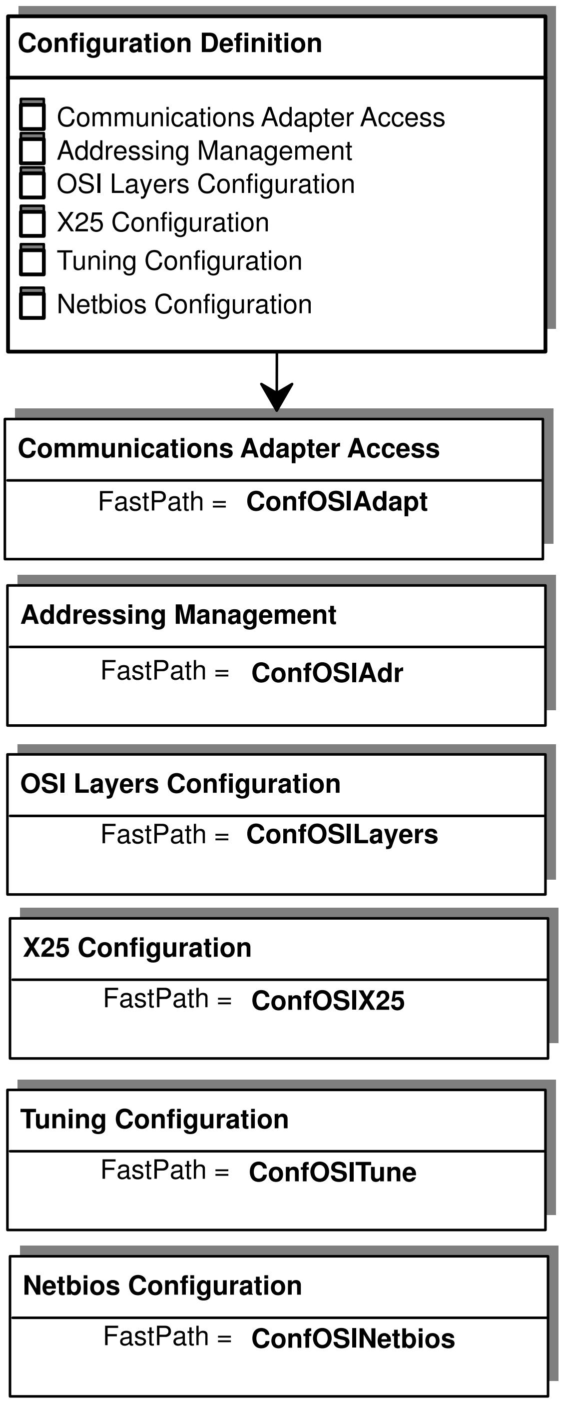


Figure 45. Configurator – Configuration Definition Menu.

Communications Adapter Access Menu

From the Configuration Definition menu, this menu allows the management of communications adapters by listing adapters in the configuration, or by adding or removing adapters to/from the configuration.

Figure 46 shows the Communications Adapter Access menu:

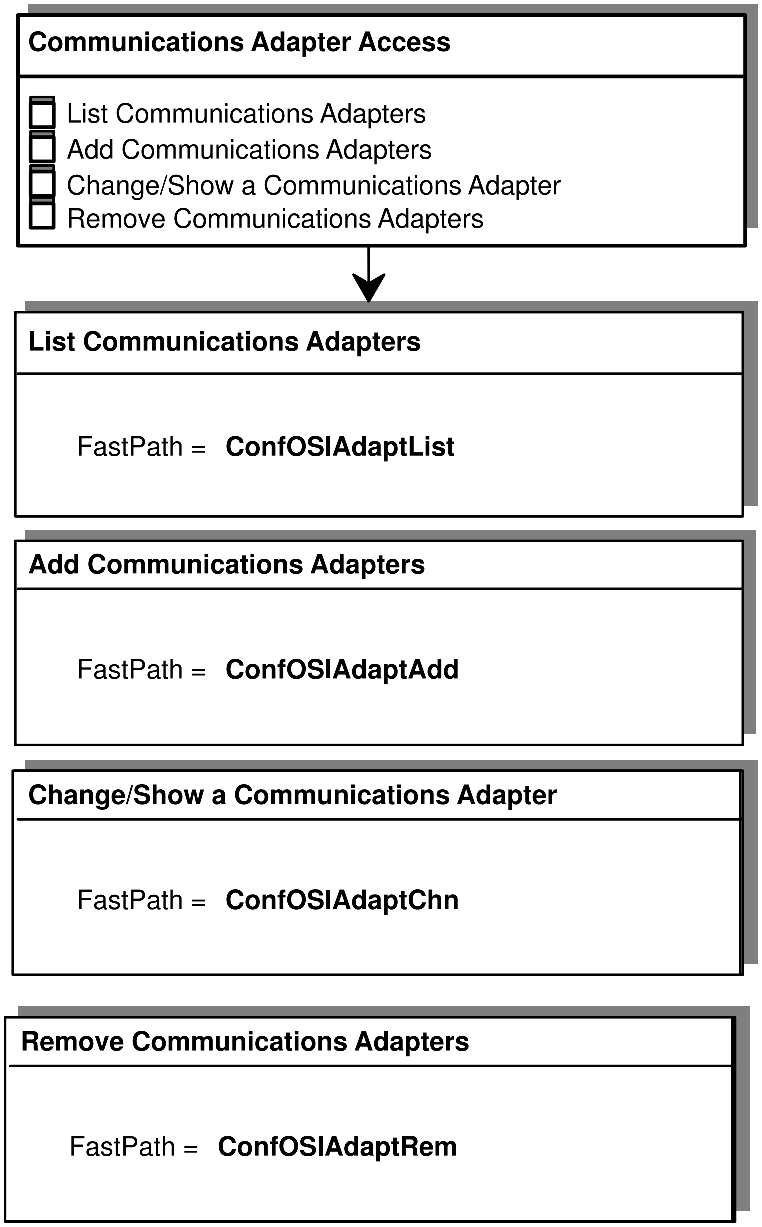


Figure 46. Configurator – Communications Adapter Access Menu.

Addressing Management Menu

From the Configuration Definition menu, this menu allows you to configure configuration addressing parameters: Subnetwork entries, local and remote NSAPs, route entries, relay route entries, profile entries and security filters.

Figure 47 shows the Addressing Management menu:

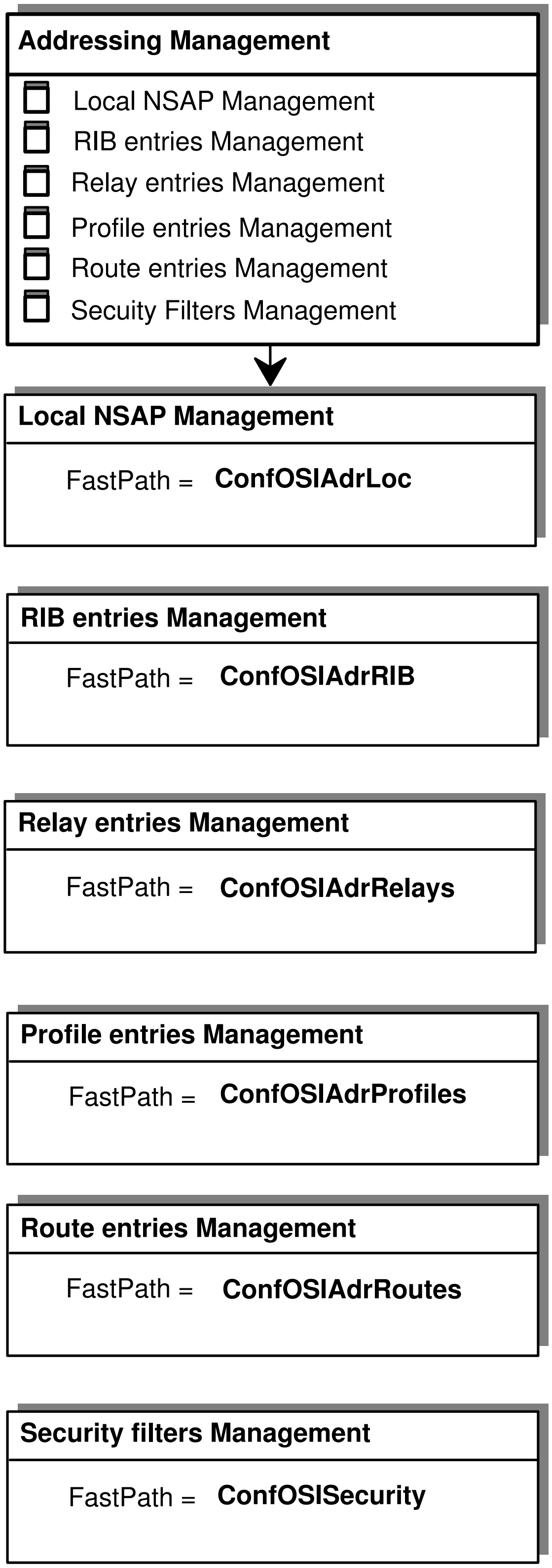


Figure 47. Configurator – Addressing Management Menu.

OSI Layers Configuration Menu

From the Configuration Definition menu, this menu allows you to configure the OSI Layers.

Figure 48 shows the OSI Layers Configuration menu:

OSI Layers Configuration
FastPath = **ConfOSILayers**

Figure 48. Configurator – OSI Layers Configuration Menu.

X.25 Configuration Menu

From the Configuration Definition menu, this menu allows you to configure an X.25 network: definition of the Subsequent Protocol Identifier (SPI) and the Permanent Virtual Circuit (PVC) tables.

Figure 49 shows the X.25 Configuration menu:

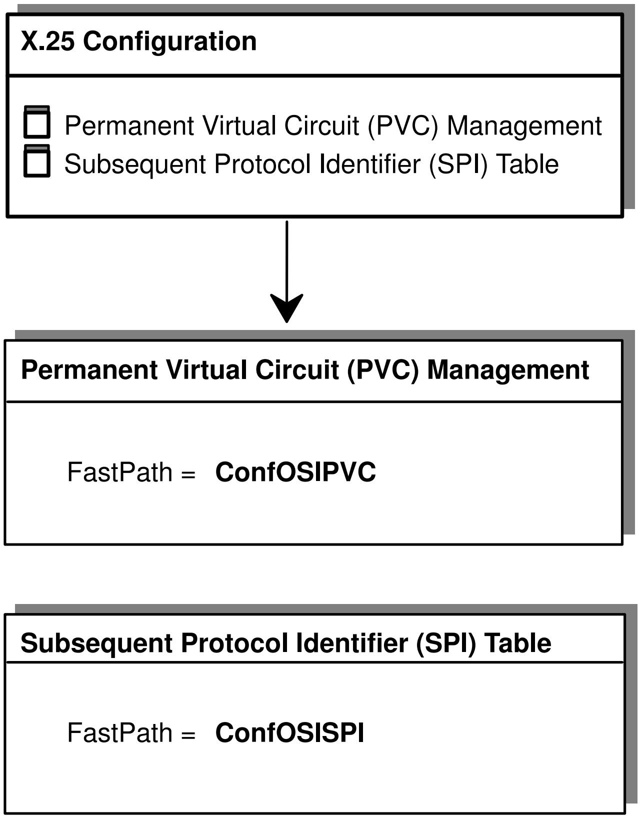


Figure 49. Configurator – X.25 Configuration Menu.

Tuning Configuration Menu

From the Configuration Definition menu, this menu allows you to tune your OSI layers configuration by setting timers, parameters and buffer memory size.

Figure 50 shows the Tuning Configuration menu:

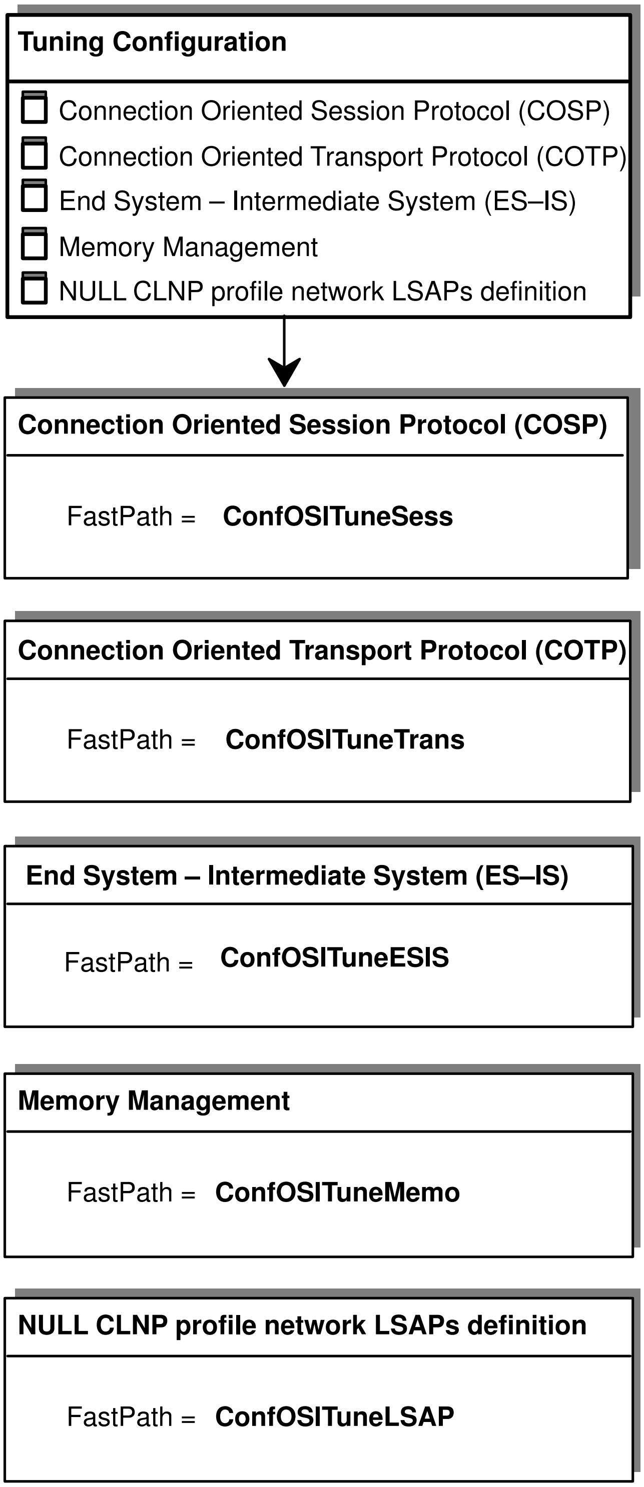


Figure 50. Configurator – Tuning Configuration Menu.

Netbios Configuration Menu

From the Configuration Definition menu, this menu allows you to show and define the Netbios configuration.

Figure 51 shows the Netbios Configuration menu:

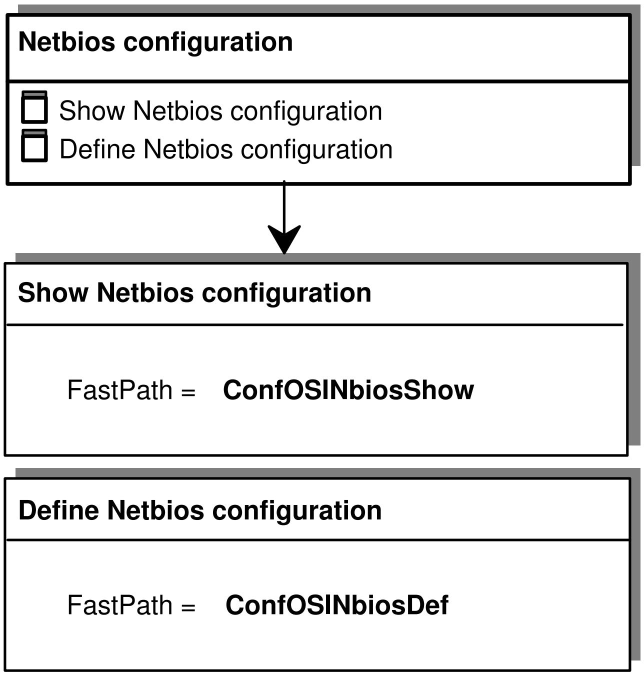


Figure 51. Configurator – Netbios Configuration Menu.

Local NSAP Management Menu

From the Addressing Management menu, this menu allows you to list, add and remove the local NSAPs.

Figure 52 shows the Local NSAP Management menu:

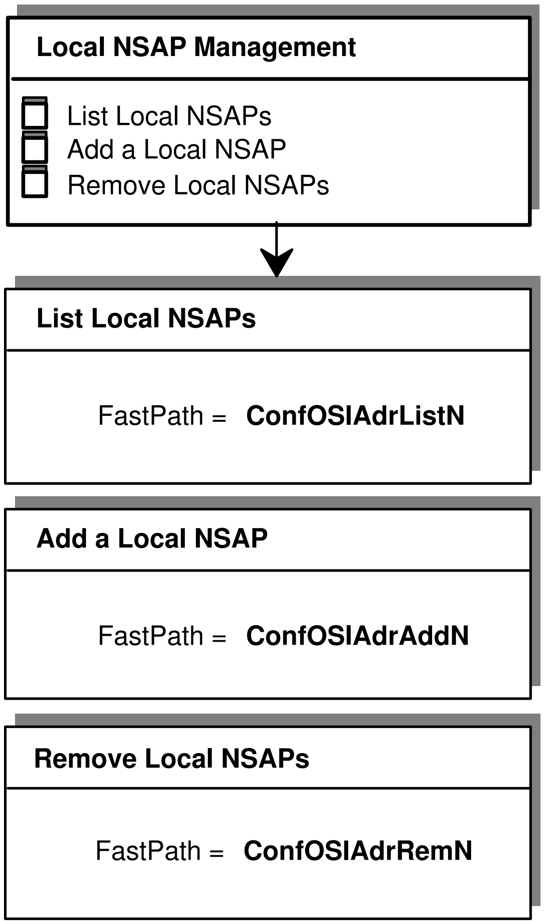


Figure 52. Configurator – Local NSAP Management Menu.

Subnetwork Entries Management Menu

From the Addressing Management menu, this menu allows you to list, add, change and remove the subnet entries.

Figure 53 shows the Subnetwork Entries Management menu:

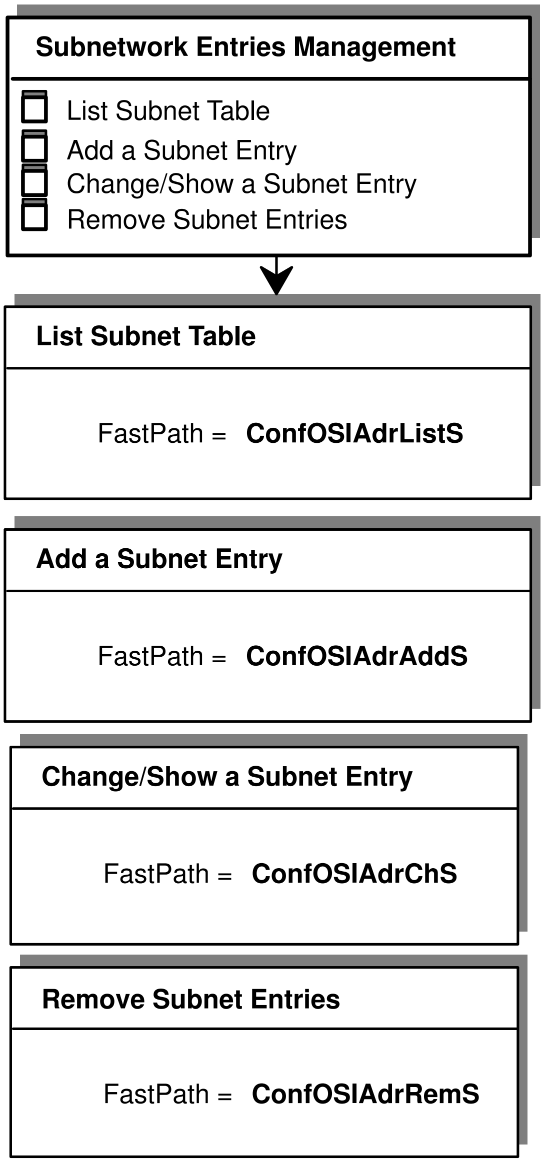


Figure 53. Configurator – Subnetwork Entries Management Menu.

RIB Entries Management Menu

From the Addressing Management menu, this menu allows you to list, add, change and remove the RIB entries.

Figure 54 shows the RIB Entries Management menu:

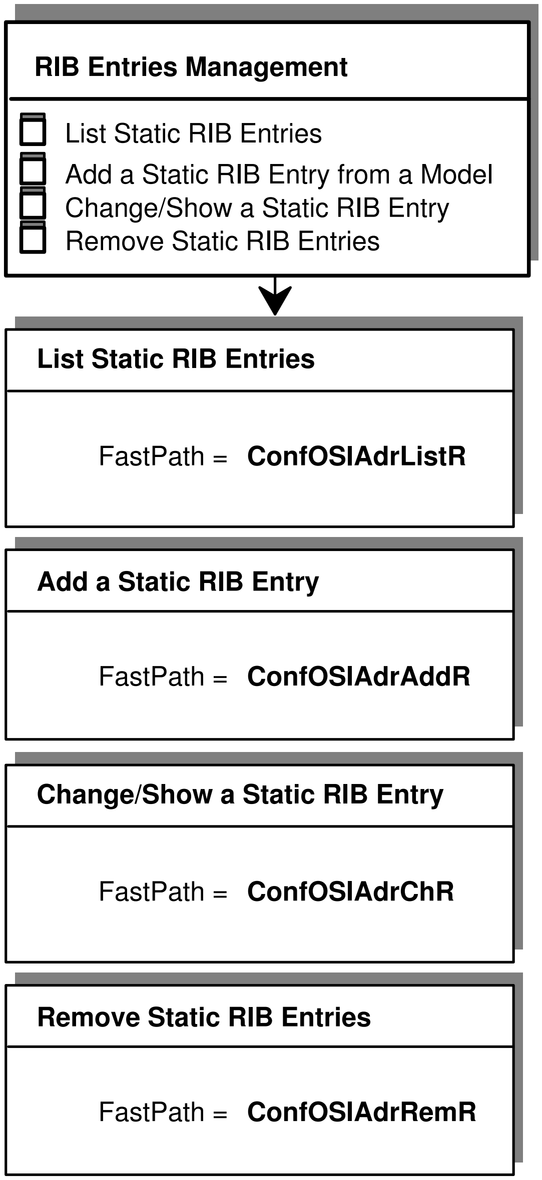


Figure 54. Configurator – RIB Entries Management Menu.

Relay Entries Management Menu

From Addressing Management menu, this menu allows you to list, add and remove the relay entries.

Figure 55 shows the Relay Entries Management menu:

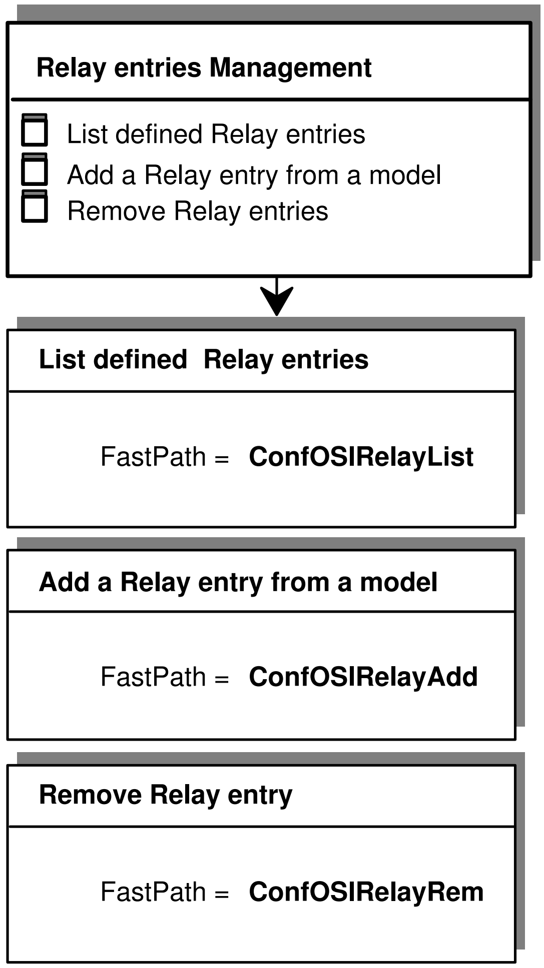


Figure 55. Configurator – Relay Entries Management Menu.

Profile Entries Management Menu

From Addressing Management menu, this menu allows you to define, add and remove profile entries.

Figure 56 shows the Profile Entries Management menu:

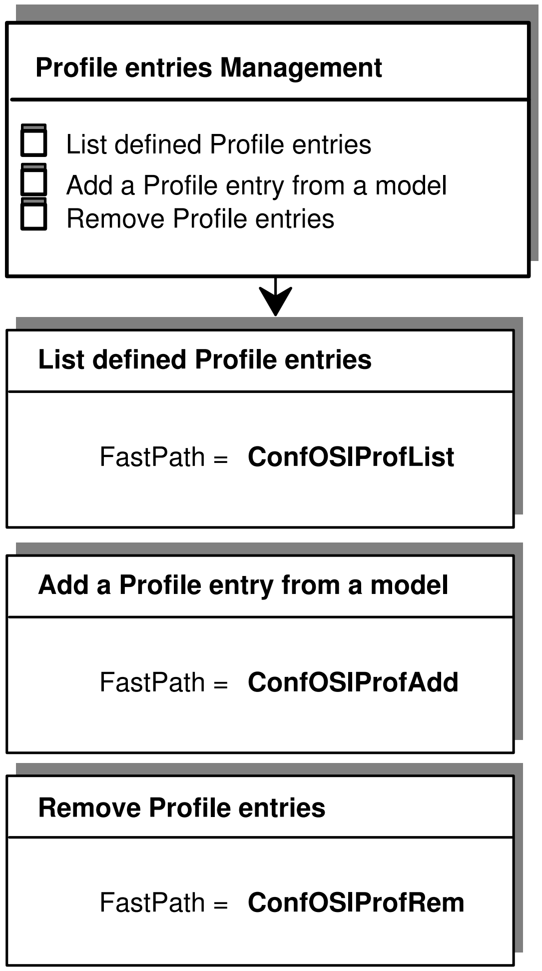


Figure 56. Configurator – Profile Entries Management Menu.

Route Entries Management Menu

From Addressing Management menu, this menu allows you to define, add and remove route entries.

Figure 57 shows the Route Entries Management menu:

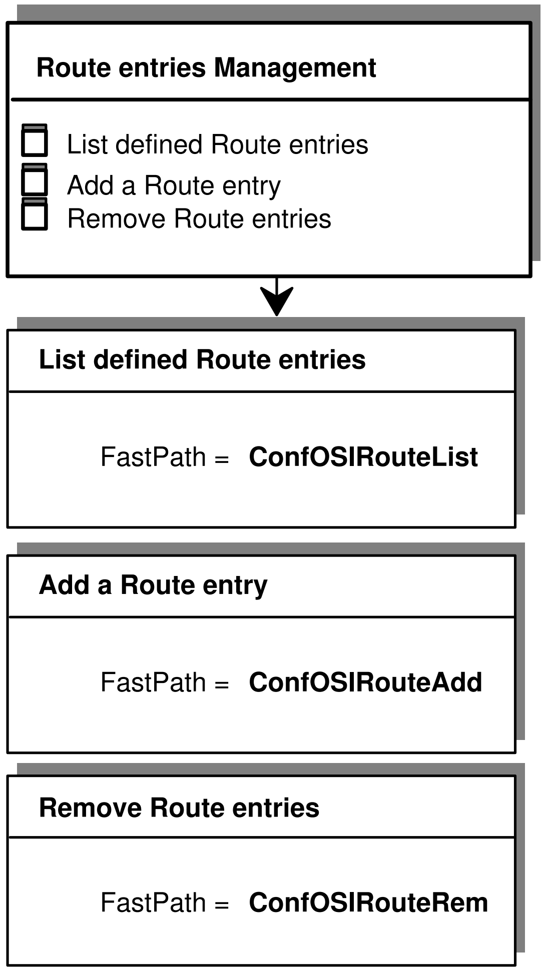


Figure 57. Configurator – Route Entries Management Menu.

Security Filter Management Menu

From Addressing Management menu, this menu allows you to change/show, list, add and remove security filter entries.

Figure 58 shows the Security Filter Management menu:

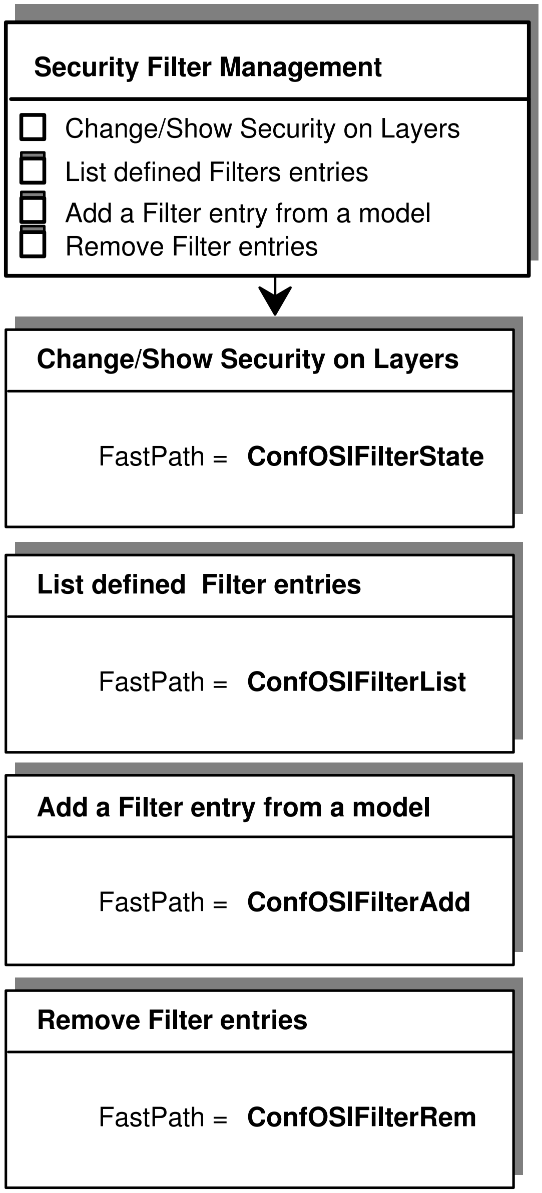


Figure 58. Configurator – Security Filter Management Menu.

Unload Current Loaded Configuration Menu

From the OSI Configuration menu, this menu de–activates the previously loaded configuration. It takes into account the stopping of services and applications known by the OSI Stack and the unloading of drivers.

Note: It does not stop other applications, not declared to the OSI Stack, using OSI services. Stopping these applications is **mandatory** in order to unload a configuration, and must be performed by the administrator.

Figure 59 shows the Unload Current Loaded Configuration menu:

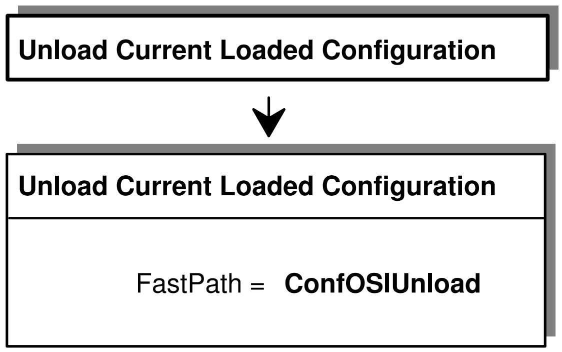


Figure 59. Configurator – Unload Current Loaded Configuration Menu.

Load the Last Selected Configuration Menu

From the OSI Configuration menu, this menu activates a defined and selected configuration. It includes the loading of the OSI driver, the initialization of OSI services and the applications which are known by the OSI Stack.

Note: The drivers of the defined adapters must be configured and in the *Available* state before loading of a configuration.

Figure 60 shows the Load the Last Selected Configuration menu:

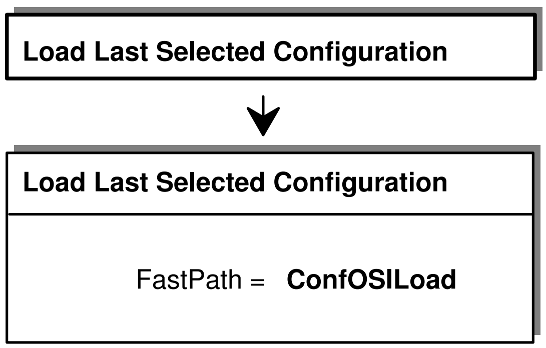


Figure 60. Configurator – Load the Last Selected Configuration Menu.

Change/Show Loaded Configuration Menu

From the OSI Configuration menu, this menu allows some dynamic–only data (dynamic RIB, waiting RIB entries, ES–IS protocol state, memory occupation) together with reset static and dynamic RIB data to be shown within an active configuration. Variations can be made to certain parameters of the current loaded configuration, for example; dynamic RIB entries, ES–IS protocol and Memory buffer status.

Figure 61 shows the Change/Show Loaded Configuration menu:

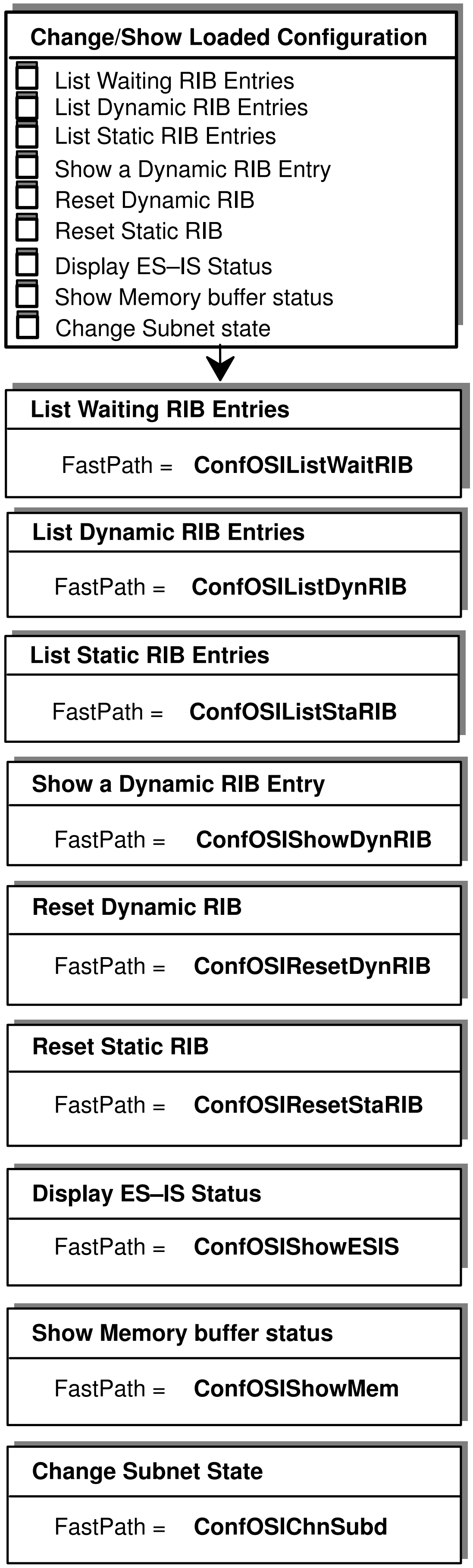


Figure 61. Configurator – Change/Show Loaded Configuration Menu.

Parameter Management – Quick Reference

The following index provides a fast method of locating detailed information about management functions.

A

C

D

E

F

H

I

P

R

S

Accessing OSI Networking

Access

From the Communications Applications & services menu, select:

OSI Networking

FastPath: `OSIstack`

Overview

This menu allows access to the OSI Stack management operations using the options shown below:

```
OSI Networking
-----------------------------------
OSI configuration
OSI Diagnosis Interactive Tool
OSI Start Applications Support
```

Description

The menu options are described below:

OSI Configuration
Manages all the configurations defined by the administrator, see page 9-32.

OSI Diagnosis Interactive Tool
Gives access to ODIT services, in the OSI Diagnostic Interactive Toolkit (ODIT) User's Guide, see Bibliography, on page B-1.

OSI Start Applications Support

Manages the OSI Stack Application database defining the application used by the SAS mechanism, see page 9-33.

Accessing OSI Configuration

Access

From the OSI Networking menu, select:

OSI Configuration

FastPath: `osi`

Overview

This menu allows access to the OSI Stack management operations using the options shown below:

OSI Configuration
Minimum Configuration Configuration Management Configuration Definition Unload Current Loaded Configuration Load the Last Selected Configuration Change/Show Loaded Configuration

Description

The menu options are described below:

Minimum Configuration
Manages all the configurations defined by the administrator, see page 9-36.

Configuration Management
Manages all the configurations defined by the administrator, see page 9-39.

Configuration Definition
Defines the parameters of the current configuration, (the last configuration selected under configuration management), see page 9-48.

Unload Current Loaded Configuration
De–activates the OSI Stack, see page 9-112.

Load the Last Selected Configuration
Activates the last loaded OSI Stack, see page 9-113.

Change/Show Loaded Configuration
Permits loaded configuration to be displayed and modified, see page 9-114.

Accessing OSI Start Applications Support

Access

From the OSI Networking menu, select:

OSI Start Applications Support

FastPath: `OSISAS`

Overview

This menu allows access to the OSI Stack Application management operations using the options shown below:

OSI Start Applications Support
List of Applications Change Applications State

Description

The menu options are described below:

List of Applications

Shows list and current state of Applications managed by SAS, see page 9-34.

Change Applications State

Allows the current state of Applications managed by SAS to be changed, see page 9-35.

How to List Applications

Access

From the OSI Start Applications Support menu, select:

List of Applications

FastPath: `OSISASList`

Command: osisas on page 11-7.

Overview

This command lists the Applications managed by SAS.

Description

The display has the following format:

```
# Application .............. State

ftam .......................  OFF
uft ..........................ON
```

Select state from ring ON / OFF.

Successful Result

The list of the available Applications and their states is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Change Applications State

Access

From the OSI Start Applications Support menu, select:

Change Applications State

FastPath: `OSISASChange`

Command: `osisas` on page 11-7.

Overview

This dialog is used to change the Application state.

All the applications known by SAS are listed, indicating their states.

```
# Application .............. State

ftam .......................  OFF
uft ..........................ON
```

Dialog Fields

***SAS Names**	For reference only. Name of applications whose states can be modified.
*** Applications State**	Select state from ring: **ON** : to validate the entry **OFF** : to invalidate the entry.

Successful Result

A message is displayed for each piece of successfully updated information.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

Accessing Minimum Configuration

Access

From the OSI Configuration menu, select:

Minimum Configuration

FastPath: `ConfOSIdefault`

Command: `OSIdefault` on page 10-11.

Overview

This menu is a ghost dialog allowing the system administrator to create and define the minimum configuration (called OSIdefault) which can then be redefined and managed like all others.

Description

The configuration OSIdefault is created, then:

all communications adapters available on the machine are added,

a subnet entry per communications adapter is defined. The first adapter with a subdomain number of 0x10, the second adapter with a subdomain number of 0x20 and so on, in increments of 0x10,

two local NSAPs (if node name and system unit hardware have been defined) are created with type Local.

Successful Result

If the OSIdefault configuration already exists, an end message is displayed.

Display of minimum configuration parameters.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

In the example the system unit hardware has the identity **rs5**; this is encoded as 72 73 35 in the local NSAP name.

See overleaf.

Example

```
================================================================================
COMMUNICATIONS STATUS:

Configuration name          : OSIdefault
       (automatically generated)
             Date of last modification   : 96–01–25 09:37
             This configuration is the Selected one.
             This configuration is the Loaded one.
================================================================================
COMMUNICATIONS ADAPTERS:            Adapter      Subdomain   State Adapter–Type
             00–00–0E       0x10       up    Integrated Ethernet Adapter
             00–02–01–00  0x20       up    Hispeed WAN Comm Adapter Line
             00–02–01–01  0x30       up    Hispeed WAN Comm Adapter Line
             00–02–01–02  0x40       detachHispeed WAN Comm Adapter Line
             00–02–01–03  0x50       detachHispeed WAN Comm Adapter Line
================================================================================
X25 CONFIGURATION:

Subsequent Protocol Identifier Table (SPI) :

             module when there is no DATA field in a packet : 0x0410
             module when unknown SPI                        : 0x3610
             *********************************
             * SPI nb *    value   *  module *
             *********************************
             *   1    * 0x00      *  0x0410 *
             *   2    * 0x01      *  0x3610 *
             *   3    * 0x02      *  0x3610 *
             *   4    * 0x03      *  0x0410 *
             *   5    * 0x81      *  0x0F10 *
             *   6    * 0x01      *  0x3610 *
             *   7    * 0x01      *  0x3610 *
             *   8    * 0x01      *  0x3610 *
             *   9    * 0x01      *  0x3610 *
             *  10    * 0x01      *  0x3610 *
             *  11    * 0x01      *  0x3610 *
             *  12    * 0x01      *  0x3610 *
             *  13    * 0x01      *  0x3610 *
             *  14    * 0x01      *  0x3610 *
             *  15    * 0x01      *  0x3610 *
             *  16    * 0x01      *  0x3610 *
             *********************************

List of PVC of the   00–02–01–00    Hispeed WAN Comm Adapter Line :

No PVCs defined for this adapter.

No X25 Facilities defined for this adapter.

List of PVC of the   00–02–01–01    Hispeed WAN Comm Adapter Line :

No PVCs defined for this adapter.

No X25 Facilities defined for this adapter.

List of PVC of the   00–02–01–02    Hispeed WAN Comm Adapter Line :

No PVCs defined for this adapter.
```

```
No X25 Facilities defined for this adapter.

List of PVC of the   00–02–01–03    Hispeed WAN Comm Adapter Line :

No PVCs defined for this adapter.

No X25 Facilities defined for this adapter.

==========================================================================
OSI LAYERS CONFIGURATION: Max. no. of simultaneous COTP connections      : 64
          Max. number of COTP connections on a VC                        : 64
          Use of the ES–IS protocol                                      : Yes
          Type of function ES–IS                                         : ES
==========================================================================
OSI TUNING CONFIGURATION:

          COTP LAN local retransmission time (T1)                        : 3000 ms
          COTP WAN local retransmission time (T1)                        : 15000 ms
          Max no. of transmissions of a COTP LAN PDU (N)                 : 10
          Max no. of transmissions of a COTP WAN PDU (N)                 : 10
          Accept Different Responding Address                  : Yes

          ES timer                                   : 180 s
          IS timer                                   : 180 s

          NULL CLNP profile Network LSAPs                   : 20

          Buffer memory size                        : automatic

          Max. no. Static RIB entries in memory          : 256
          Max. no. Static–Fix RIB entries in memory      : 200
==========================================================================
NETBIOS PARAMETERS:
          Netbios Subnet numbe      : 0x0
          Netbios NSAP              : 0x49000009006a00010001
          Netbios LSAP              : ec

==========================================================================
ADDRESSING CONFIGURATION:

          no. of defined Profiles       : 0
          no. of defined Routes         : 0
          no. of defined Relays         : 0
          no. of defined Filters        : 0
          Security on COTP:CLTP:CLNP     : 0:0:0

          List of the Local NSAPs:
          0x4972733701                  : LOCAL
          0x490022407701                : LOCAL
          0x497273370101                : LOCAL
          0x497273370102                : GROUP

No RIB entries defined.
==========================================================================
```

Note: In this example, the Netbios Subdomain number is initialized to a value of 0 (zero). At loading time this value is changed to the lowest Ethernet number defined in the configuration.

Accessing Configuration Management

Access

From the OSI Configuration menu, select:

Configuration Management

FastPath: `ConfOSIMag`

Overview

This menu allows the system administrator to manage all the defined configurations using the options shown below:

Note: Super–user authority is required for all except ”List” and ”Show” Configurations.

```
Configuration Management
----------------------------------------
List all Configurations
Create a Configuration
Remove a Configuration
Select a Configuration
Show a Configuration
Change Configuration Name or Comment
```

Description

The menu options are described below:

List all Configurations
Lists all the configurations, sorted in creation order, on page 9-40.

Create a Configuration
Creates a configuration with default values, or a copy of another configuration, on page 9-41.

Remove a Configuration
Deletes one or more configurations from those created, on page 9-43.

Select a Configuration
Selects the current configuration, on page 9-44.

Show a Configuration
Displays the detailed contents of a created configuration, on page 9-45.

Change Configuration Name or Comment
Allows the name or comment of a configuration to be modified, on page 9-46.

How to List all Configurations

Access

From the Configuration Management menu, select:

List all Configurations

FastPath: `ConfOSIList`

Command: `osiconf` on page 10-4.

Overview

The output window lists all configurations stored in the OSI configuration data base. The selected, generated and loaded configurations, if they exist, are indicated.

Description

The result is formatted in several columns:

```
<conf_name>   <date/time>   <comment>   [Sel | Load]
```

where:

Configuration Name Name in10 characters maximum, without space.

Date/Time Date and time of the last modification.

Comment Creator’s remark.

Sel **Selected flag**: indicates the current configuration, that is the configuration selected with the menu How to Select a Configuration on page 9-44.

Load **Loaded flag**: indicates the current loaded active configuration.

Successful Result

The command displays a list of the defined configurations or a message saying that the list is empty.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
NAME        DATE              COMMENT                 STATE
conf        92-10-08 10:13    configuration example   Sel   Load
confex      92-10-08 10:13
```

How to Create a Configuration

Access

From the Configuration Management menu, select:

Create a Configuration

FastPath: `ConfOSICreate`

Command: `osiconf` on page 10-4.

Overview

This dialog is used to create new configurations. These new configurations are initialized either from the current selected one, or, with default values stored in the "basic configuration".

The basic configuration is a virtual configuration whose values are the default ones. It is unlike the other configurations in that it is never listed or duplexed and cannot be managed or removed.

The new configuration is selected automatically and becomes the current configuration. It is stored in the OSI configuration data base.

To create a new configuration:

- Copy the basic configuration containing some default values

  or

- Copy another configuration already defined.

In the case of an already defined configuration, the new configuration is identical apart from the name, comment and the date. Once the creation is made, the two configurations (the new one and the one from which it was created), are completely independent. When one is modified, the other remains unchanged.

Dialog Fields

***Creation mode**	Choose one of the following from the list: **Basic configuration** **From the last selected configuration** (the current one). Default value: can be one or other from list.
***Configuration name**	10 characters maximum. Alphanumeric character string and the following '_' '.' '/' and '\'. Duplication of the configuration name is not authorized.
Configuration Comment	40 characters maximum. Alphanumeric character string and the following '_' '.' '/' and '\'.

Successful Result

A confirmation message is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
The configuration confex has been created
```

How to Remove a Configuration

Access

From the Configuration Management menu, select:

Remove a Configuration

FastPath: `ConfOSIRem`

Command: `osiconf` on page 10-4.

Overview

This dialog is used to remove a configuration from the OSI configuration data base.

To delete a configuration, select one configuration name from the list and confirm the selection.

Note: The loaded configuration cannot be deleted.
Only one configuration can be deleted at a time.

Dialog Fields

Name of the configuration	All the created configurations are listed in alphabetical order. The state (*Selected, Loaded*) of each configuration is indicated.

Successful Result

The selected configuration is removed and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
The configuration confex1 has been deleted
```

How to Select a Configuration

Access

From the Configuration Management menu, select:

Select a Configuration

FastPath: `ConfOSISel`

Command: `osiconf` on page 10-4.

Overview

This dialog is used to select the current configuration. The selected configuration is identified by the flag **sel** in the List menu.

The current configuration does not need to be defined at each activation of the configurator; the last–selected configuration is saved from one activation to another.

To select a configuration, select one configuration name from the list and confirm the selection.

Note: All information entered from the Configuration Definition menu on page 9-48 concerns this selected configuration.

Dialog Fields

***Name of the configuration**	All the created configurations are listed in alphabetical order. The state (*Selected, Loaded*) of each configuration is indicated.

Successful Result

A confirmation message displays the name of the selected configuration.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
The configuration conf has been selected
```

How to Show a Configuration

Access

From the Configuration Management menu, select:

Show a Configuration

FastPath: `ConfOSIDisp`

Command: `osiconf` on page 10-4.

Overview

This dialog displays all the information concerning a given configuration. This is the only access to detailed configuration characteristics.

To display a configuration, select a name from the list of the contents of the data base and confirm the selection.

Dialog Fields

***Name of the configuration** All the created configurations are listed in alphabetical order. The state (*Selected, Loaded*) of each configuration is indicated.

The characteristics of the selected configuration are displayed in the output in the following manner:

- Configuration status: name, comment, date and status
- Communications adapter access, sorted by type
- X.25 configuration information (if X.25 is used) PVCs, SPI table and Facilities
- OSI layers configuration
- OSI Tuning configuration
- OSI Multicast for Netbios
- Addressing configuration:
  - Local NSAPs
  - RIB entries
  - profile entries
  - route entries
  - transport relay entries
  - security entries

  These last three entries are sorted in creation order – the order in which objects have been added to the ODM. This only applies if the lower layer package is installed.

Successful Result

A detailed description of the configuration is displayed. The information is structured in sections, identified by an upper–case title and separated by horizontal lines.

WARNINGS: None.

Error messages: None.

EXAMPLE:

See OSIdefault Command on page 10-11.

How to Change a Configuration Name or Comment

Access

From the Configuration Management menu, select:

Change a Configuration Name or Comment

FastPath: `ConfOSIChange`

Command: `osiconf` on page 10-4.

Overview

This dialog is used to change the configuration name, the comment or both.

The name of a configuration must remain unique.

A list displays the available configurations allows the selection of the configuration to be changed.

Note: The name of the loaded configuration cannot be changed.

Dialog Fields

***Name of configuration**	All the created configurations are listed in alphabetical order. The state (*Selected, Loaded*) of each configuration is indicated.
Old configuration name	For reference only.
New configuration name	Must be unique and 10 characters maximum. Alphanumeric character string including only '_' '.' '/' and '\'. The old configuration name is given by default.
New comment	40 characters maximum. Alphanumeric character string including only '_' '.' '/' and '\'. The old configuration comment is given by default.

Successful Result

A message is displayed for each piece of successfully updated information.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

Name change only:

```
The name of the configuration conf has been modified to config
```

Comment change only:

```
The comment of the configuration config has been modified
```

Both name and comment changed:

```
The name of the configuration config has been modified to conf
The comment of the configuration config has been modified
```

Accessing Configuration Definition

Access

From the OSI Configuration menu, select:

Configuration Definition

FastPath: `ConfOSIDef`

Overview

This menu allows the system administrator to define the selected configuration. All commands stemming from this menu will affect the current configuration, the last to have been selected. When the loaded configuration is selected, the commands affect parameters in the database and memory. Menu options are shown below:

```
Configuration Definition
-------------------------
communications adapter Access
Addressing Management
OSI Layers Configuration
X25 Configuration
Tuning Configuration
Netbios Configuration
```

Description

The menu options are described below:

Communications Adapter Access

Definition of communications adapters configuration on page 9-49.

Addressing Management

Management of local NSAPs, Subnetwork and RIB entries on page 9-59.

OSI Layers Configuration

Configuration of OSI Stack parameters on page 9-93.

X25 Configuration

Configuration of X.25 Communications Adapter on page 9-95.

Tuning Configuration

Management of tuning parameters on page 9-103.

Netbios Configuration

Definition of multicast address for Netbios on page 9-109.

Accessing Communications Adapters

Access

From the Configuration Definition menu, select:

Communications adapter access

FastPath: `ConfOSIAdapt`

Overview

This menu allows the system administrator to define which communications adapters are configured, according to the hardware installed on the machine, as shown below:

```
Communications adapter access
-----------------------------
List communications adapters
Add communications adapters
Change/Show communications adapters
Remove communications adapters
```

Description

The menu options are described below:

List communications adapters

Lists configured communications adapters on page 9-50.

Add communications adapters

Definition of adapter codes that can be configured for each type of adapter on page 9-52.

Change/Show communications adapters

Displays and enables modification of adapter attributes already defined in a configuration on page 9-54.

Remove communications adapters and dependencies

Removal of communications adapters on page 9-57.

How to List Communications Adapters

Access

From the Communications adapter Access menu, select:

List Communications Adapters

FastPath: `ConfOSIAdaptList`

Command: osiadapt on page 10-2.

Overview

This command lists the network adapter type and Adapter Location Code (ALC), for each communications adapter, see Adapter Location Codes on page 2-2.

Note: A configuration must already be selected, see How to Select a Configuration on page 9-44.

Description

The display has the following format:

```
<adapter_location code> <subdomain>  <state>  <adapter_type>
```

Entries are presented by type.

Adapter Type

Ethernet High–Performance LAN Adapter

Standard Ethernet Adapter

HiSpeed WAN Comm. Adapter (single & 4–port)

Token Ring High–Performance Network Adapter

Fiber Distributed Data Interface (FDDI) Network Adapter (All adapters)

Adapter Location Code

The location code for the Standard Ethernet Adapter is as follows:

00–00–0E (where E is the connector number on the CPU board).

The location codes for the Ethernet LAN, the Token Ring and the FDDI Adapters are as follows:

00–03 (where 3 is the slot number),

The location codes for the HiSpeed WAN Comm. Adapter are as follows:

00–06–01–02 (where 6 is the slot number and 2 the line number)

Subdomain

The number identifying the subdomain associated with the communications adapter is as folllows:

10

State

The state of each communications adapter is as follows:

UP or DETACH

Successful Result

The list of the communications adapters with network types and associated adapter codes, states and subdomains, is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
LOCATION        SUBDOMAIN       STATE   ADAPTER TYPE
00-00-0E        10              UP      Standard Ethernet Adapter
00-06-01-02     50              DETACH  HiSpeed WAN Comm. Adapter
```

How to Add Communications Adapters

Access

From the Communications Adapter Access menu, select:

Add communications adapters

FastPath: `ConfOSIAdaptAdd`

Command: osiadapt on page 10-2, and osix25 on page 10-36.

Overview

This menu allows the definition of the adapter to be configured as one of the communications adapter types.

A list displays the *Available* communications adapters.

One location code may be selected from the list. A location code identifies a network attachment.

A configuration must already be selected, see How to Select a Configuration on page 9-44.

The command will be effective in memory (dynamic) if the configuration is the loaded one.

Dialog Fields

***Adapter Code** Location code of the communications adapter to be defined. It is selected from the list.

***State** Initial state of the communications adapter to be configured: **up** or **detach**. Default is **up**.

Action when the selected configuration is not loaded:
– If the state is **up**, then this adapter is activated (attached) when the configuration is loaded. If the activation fails, the state is set to **detach** in the data base.

Action when the selected configuration is loaded:
– If the state is **up**, then this adapter is activated (attached) immediately after its successful configuration. If the activation fails, the adapter is not configured in the data base.

For LAN only:

Maximum NPDU size sent Appears only for LAN adapters. It defines the maximum size of the NPDU frame sent by the ConnectionLess Network Protocol (CLNP). The size depends on the medium:
On Ethernet : a decimal value in the range from **60** to **1500** . Default is **1500**.
On Token Ring and FDDI: a decimal value in the range from **60** to **4000**. Default is **4000**.

For WAN only (Added Facilities):

Negotiation of the packet size The facility ”negotiation of the packet size” may be forced in the call request packet if this field is set to **Yes**. Default is **No**.

Negotiation of the window size The facility ”negotiation of the window size” may be forced in the call request packet if this field is set to Yes. Default is **No**.

Negotiation of the throughput class
The facility ”negotiation of the throughput class” is forced in the call request packet if this field is set to **Yes**. Default is **No**.

Negotiation of the minimum throughput class
The facility ”negotiation of the minimum throughput class” may be forced in the call request packet if this field is set to **Yes**. Default is **No**.

Reverse Charging Request
The facility ”Reverse Charging Request” may be forced in the call request packet if this field is set to **Yes**. Default is **No**.

Fast Select
The facility ”Fast Select” may be forced in the call packet to the value selected in the ring: **not expected**, **expected without restriction on the reply**, **expected with restriction on the reply**. Default is **not expected**.

Transit Delay Value
The facility ”transit delay value” may be forced in the call request packet to an integer value in the range from **0** to **65534**. **65335** means no transit delay defined. The other values are timer specified in milliseconds. Default is **65535**.

End to End Transit Delay Value (in CALL REQUEST packet)
The facility ”end to end transit delay value” may be forced in the call request packet to an integer value in the range from **0** to **65534**. **65335** means no end to end transit delay defined. The other values are timer specified in milliseconds. Default is **65535**.

End to End Transit Delay Value (in CALL CONFIRM packet)
The facility ”end to end transit delay value” may be forced in the call confirm packet to an integer value in the range from **0** to **65534**. **65335** means no end to end transit delay defined. The other values are timer specified in milliseconds. Default is **65535**.

Successful Result

The name of the changed configuration is displayed, followed by the name of the adapter which has been successfully added.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

The communications adapter Standard Ethernet Adapter 00-01-0E has
been added.
```

CAUTION

A network attachment is identified by its location code. If an adapter must be moved, the attachments defined by the old location code must be removed and new ones created.

If a communications adapter is removed from the station, the configurations are not automatically updated.

How to Change/Show Communications Adapters

Access

From the Configuration Definition menu, select:

Communications adapter access

FastPath: `ConfOSIAdaptChn`

Command: osiadapt on page 10-2 and osix25 on page 10-36.

Overview

This command displays and enables modification of the following attributes of a communications adapter already defined in a configuration:

- the adapter code (loaction),
- the subdomain number,
- the state of the adapter,
- for a LAN adapter: the maximum NPDU size sent,
- for a WAN adapter: the facilities which can be forced in an outgoing call request or call confirm packet.

All the defaults values are those already configured, or the defaults used to add a communications adapter.

The configuration has to be already selected.

If the selected configuration is loaded, the memory is updated.

Note: When changing any attribute of the communications adapter, if the configuration is loaded and the adapter is up, the established connections on it can be broken.

Dialog Fields

***New Adapter Code** The displayed value is the Location code of the communications adapter already defined.
If an adapter is moved to a new location, it is necessary to change its adapter code. A list details only the adapters of the same type installed and available in the machine. The new adapter code must not be already defined in the configuration. Select the location code from this list. If the configuration is loaded, the communications adapter is de-activated (detached) before its code is modified and if the state is up, the new adapter is activated as for an ”Add communications adapter”. The objects depending on the old adapter code (X.25 facilities and PVCs) are updated.

Warning: When de-activating a communications adapter, the established connections on its subdomain can be broken.

***State** Initial state of the communications adapter to be configured: **up** or **detach**. Default is **up**.

Action when the selected configuration is not loaded:
– If the state is **up**, then this adapter is activated (attached) when the configuration is loaded. If the activation fails, the state is set to **detach** in the data base.

Action when the selected configuration is loaded:
– If the state is **up,** then this adapter is activated (attached) immediately after its successful configuration. If the activation fails, the adapter is not configured in the data base.

For LAN only:

Maximum NPDU size sent
Appears only for LAN adapters. It defines the maximum size of the NPDU frame sent by the ConnectionLess Network Protocol (CLNP). The size depends on the medium:
On Ethernet : a decimal value in the range from **60** to **1500** . Default is 1500.
On Token Ring and FDDI: a decimal value in the range from **60** to **4000**. Default is 4000.

For WAN only (Added Facilities):

Negotiation of the packet size
The facility ”negotiation of the packet size” may be forced in the call request packet if this field is set to **Yes**. Default is **No**.

Negotiation of the window size
The facility ”negotiation of the window size” may be forced in the call request packet if this field is set to Yes. Default is **No**.

Negotiation of the throughput class
The facility ”negotiation of the throughput class” is forced in the call request packet if this field is set to **Yes**. Default is **No**.

Negotiation of the minimum throughput class
The facility ”negotiation of the minimum throughput class” may be forced in the call request packet if this field is set to **Yes**. Default is **No**.

Reverse Charging Request
The facility ”Reverse Charging Request” may be forced in the call request packet if this field is set to **Yes**. Default is **No**.

Fast Select
The facility ”Fast Select” may be forced in the call packet to the value selected in the ring: **not expected**, **expected without restriction on the reply**, **expected with restriction on the reply**. Default is **not expected**.

Transit Delay Value
The facility ”transit delay value” may be forced in the call request packet to an integer value in the range from **0** to **65534**. **65335** means no transit delay defined. The other values are timer specified in milliseconds. Default is **65535**.

End to End Transit Delay Value (in CALL REQUEST packet)
The facility ”end to end transit delay value” may be forced in the call request packet to an integer value in the range from **0** to **65534**. **65335** means no end to end transit delay defined. The other values are timer specified in milliseconds. Default is **65535**.

End to End Transit Delay Value (in CALL CONFIRM packet)
The facility ”end to end transit delay value” may be forced in the call confirm packet to an integer value in the range from **0** to **65534**. **65335** means no end to end

transit delay defined. The other values are timer specified in milliseconds. Default is **65535**.

Successful Result

A confirmation message is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Remove Communications Adapters and Dependencies

Access

From the Communications adapter Access menu, select:

Remove Communications Adapters

FastPath: `ConfOSIAdaptRem`

Command: osiadapt on page 10-2.

Overview

This menu allows the deletion of communications adapters from the configuration list.

Select one or more adapters from the list and confirm the selection.

A configuration must already be selected, see How to Select a Configuration on page 9-44.

If the Subnet entry is removed before the associated communications adapter, the RIB entries will be kept in the database. They can be used for another communications adapter by the definition or modification of a Subnet entry, or by the modification of the RIB entries.

If the configuration is the loaded one, the command is also performed on the memory and in that case the connections coming through this adpater are rejected/discarded.

Dialog Fields

***Adapter Code** Code of the selected adapter among the listed configured communications adapters (showing adapter type and adapter code, sorted by type).

***Remove also associated objects (RIB, Relays) ?**

A ring proposes the choice **Yes, No**. If the choice is Yes, all dependencies of the communications adapter are also removed:
– the Subnet entry associated,
– the RIB entries associated to the Subnet entry,
and if the communications adapter is an X25 type:
– the PVCs defined,
– the X25 facilities.
Defaults is No.

Successful Result

The name of the configuration is displayed, followed by the name of the adapter which has been successfully removed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf
The communications adapter 00-01-0E has been removed.
The communications adapter 00-06-01-02 has been removed.
```

Accessing Addressing Management

Access

From the Configuration Definition menu, select:

Addressing Management

FastPath: `ConfOSIAdr`

Overview

This menu allows the system administrator to manage addressing objects defined below:

```
Addressing Management
Local NSAPs Management
RIB entries Management
Relay entries Management
Profile entries Management
Route entries Management
Security Filters Management
```

Description

The menu options are described below:

Local NSAPs Management

Manages local NSAPs on page 9-59.

RIB entries Management

Manages RIB entries on page 9-63.

Relay entries Management

Manages relay entries on page 9-71.

Profile entries Management

Manages profile entries on page 9-77.

Route entries Management

Manages route entries on page 9-82.

Security Filters Management

Manages security filters entries on page 9-86.

Accessing Local NSAP Management

Access

From the Addressing Management menu, select:

Local NSAPs Management

FastPath: `ConfOSIAdrLoc`

Overview

This menu allows the system administrator to manage addressing objects defined below:

```
Addressing Management
-------------------------
List local NSAPs
Add a local NSAP
Remove local NSAPs
```

Description

The menu options are described below:

List local NSAPs

Lists local NSAPs in creation order on page 9-60.

Add a local NSAP

Adds local NSAP to list on page 9-61.

Remove local NSAPs

Removes local NSAP from list on page 9-62.

How to All List Local NSAPs

Access

From the Local NSAPs Management menu, select:

List all Local NSAPs

FastPath: `ConfOSIAdrListN`

Command: `osinsap` on page 10-18

Overview

This command lists all the local NSAPs in creation order (the order in which objects have been added to the Database).

Description

The defined NSAPs are listed with a double column format:

```
<NSAP name>  <NSAP type>
```

NSAP name The local NSAP name is shown in an even number of hexadecimal characters (2 minimum, 40 maximum).

NSAP type Type of the NSAP, which can be Local or Group. An NSAP whose type is 'Local' identifies the station by ES–IS protocol. An NSAP whose type is 'Group' does not.

Successful Result

The configuration name is displayed followed by a list, or a message if the list is empty.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf
  NSAP NAME
– – – – – – – – – – – – – – – – – – – – – – –
0x4972733201                    LOCAL
0x490077593501                  GROUP
```

How to Add a Local NSAP

Access

From the Local NSAPs Management menu, select:

Add a Local NSAP

FastPath: `ConfOSIAdrAddN`

Command: `osinsap` on page 10-18.

Overview

This command allows the addition of Local NSAPs. The addition is dynamically done if the selected configuration is the loaded one. The total number of NSAPs is limited to 32. No duplication is authorized.

Dialog Fields

***Local NSAP name**	Enter an even number of hexadecimal characters (2 minimum, 40 maximum). A list gives some default values which can be selected.
***NSAP type**	Enter **LOCAL** or **GROUP**: LOCAL if the NSAP is used as local station, GROUP if it is used as a Group and not used by ES–IS protocol.

Notes:

1. If the selected configuration is loaded, this parameter will be updated in memory.
2. In order for ES–IS to be ON, at least one NSAP must be defined.
3. There is no relation between local NSAP and communications adapters.

Successful Result

A local NSAP is loaded and a message of confirmation name is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf
Update in Loaded Configuration:
Local NSAP 44010204 has been added in memoru.
The local NSAP 44010204 has been added
```

How to Remove Local NSAPs

Access

From the Local NSAPs Management menu, select:

Remove Local NSAPs

FastPath: `ConfOSIAdrRemN`

Command: `osinsap` on page 10-18.

Overview

This command allows the removal of local NSAPs.

The removal is dynamically done if the selected configuration is the loaded one.

Ghost Dialog Field

Local NSAP name	Select from list of all defined local NSAPs. Multiple selection is permitted. Confirm the selection.

Successful Result

The local NSAPs are deleted and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

The local NSAP 55010204 has been removed.

Update in Loaded Configuration:

The local NSAP BA0B0666FF has been removed.

Update in Loaded Configuration:
```

Accessing RIB Entries Management

Access

From the Configuration Definition menu, select:

Accessing RIB entries Management

FastPath: `ConfOSIAdrRIB`

Overview

This menu allows the system administrator to define which communications adapters are configured, according to the hardware installed on the machine, as shown below:

RIB entries Management
List Static RIB entries Add a Static RIB entry from a model Change / Show a Static RIB entry Remove Static RIB entries

Description

The menu options are described below:

List Static RIB entries

Lists static RIB entries on page 9-64.

Add a Static RIB entry

Definition of static RIB entries on page 9-66.

Change / Show a Static RIB entry

Modify static RIB entries on page 9-69.

Remove Static RIB entries

Removal of static RIB entries on page 9-70.

How to List Static RIB Entries

Access

From the RIB entries Management menu, select:

List Static RIB entries

FastPath: `ConfOSIAdrListR`

Command: `osirib` on page 10-26.

Overview

This command allows the contents of the static RIB to be listed, in creation order.

For each RIB entry, the following information is displayed.

Description

RIB entries are displayed in the following format. See Add a Static RIB Entry on page 9-66 for field details.

```
Configuration name:  <conf_name>

Internal Identifier             : <internal identifier>
NSAP name                       : <NSAP name>
NSAP type                       : <NSAP type>
NSAP mask                       : <NSAP mask>
Network type                    : <type of network>
Connection type                 : <Connection type>
Remote SNPA type                : <Remote SNPA type>
Remote SNPA                     : <Remote SNPA >
Subdomain Number                : <subdomain number>
NSAP priority                   : <NSAP priority>
RIB part                        : <Rib part>
State                           : <State>
Message lifetime                : <Message lifetime>
CUG                             : <CUG>

Internal Identifier   ...........
```

EXAMPLE:

```
Configuration name  :  conf
        Internal Identifier  : 1
        NSAP name            : 0xFA01020304
        NSAP type            : NSAP
        NSAP mask            : 0xFFFFFFFFFF
        Network type         : ETHERNET
        Connection type      : PVC
        Remote SNPA type     : NET
        Remote SNPA          : 0x010203
        Subdomain number     : 0x100
        NSAP priority        : 20
        RIB part             : STATICV
        State                : ENABLE
        Message lifetime     : 15
        CUG                  : 0
        Internal Identifier  : 2
        NSAP name            : 0xDADA0102
        NSAP type            : NSAP
        NSAP mask            :
        Network type         : X25
        Connection type      : PVC
        Remote SNPA type     : MASK
        Remote SNPA          : 0xFFFFFFFF
        Subdomain number     : 0x200
        NSAP priority        : 12
        RIB part             : STATICV
        State                : ENABLE
        Message lifetime     : 15
        CUG                  : 0
```

How to Add a Static RIB Entry From a Model

Access

From the RIB entries Management menu, select:

Add a Static RIB entry from a Model

FastPath: `ConfOSIAdrAddR`

Command: `osirib` on page 10-26.

Overview

This command allows definition of the Static RIB of the current configuration.

At the first addition of a RIB entry, the default entry is displayed. For subsequent additions of RIB entries, the list of RIB entries is displayed. Select from list.

If the configuration is the loaded one, the memory is updated.

Description

See Addressing Configuration – Static Routing Information Base, on page 8-16.

Dialog Fields

Note: Example of Dialog Fields shown on page 9-65.

Internal Identifier — A decimal number, uniquely identifying each RIB entry.

***Remote NSAP name** — Name in an even number of hexadecimal characters (minimum 2, maximum 40).

***NSAP type** — Used for relaying purposes. A ring allows the selection of **NSAP** or **MSDSG**
Where: **NSAP** = CLNP relay. **MSDSG** = **MSDSG relay.**

Default = NSAP.

***NSAP mask** — Used in the case of global NSAP declaration (to gain RIB space in the case where several NSAPs can be configured with the same entry). A chain of hexadecimal characters indicating the significant part to be used in the previous NSAP entry for identifying the relevant NSAPs. Only 0, f, F hexadecimal characters are accepted. The length of the mask must be equal to the length of the NSAP entry (40 characters maximum). The NSAP name must contain zeros for each 0 character of the MASK.

***Network Type** — A ring allows the selection of the type of outgoing subnetwork to be used (in accordance with the subnet entries of the outgoing line to be used):
Ethernet
Token Ring
X25
FDDI.

Default = X25.

Note: In this case, "Network Type" must not be confused with the dynamic selection of profiles; it indicates the nature of the network in order to check the syntax of the *Remote SNPA* fields.

***Connection type (X25 only)** — A ring allows the selection of the type of virtual circuit to be used:

PVC
SVC.

Default = SVC.

***Remote SNPA type**

Type of Subnetwork Point of Attachment. A ring allows the selection of the remote SNPA type (the next routing address to be reached):
NET
MASK
SNPA.

MASK: Physical address of the next system to be reached is enclosed in the NSAP name. The address is retrieved by using a logical AND between the NSAP chain and the mask entered in the next *Remote NSAP* field.

NET: Address entered in the next *Remote NSAP* field is a relay–NSAP address (NSAP of an Intermediate system). The NET information is used to transfer some entry information on the entries of the relay–system in order to minimize the RIB involvement.

SNPA: Address of the next system to be reached is entered in the next *Remote SNPA* field.

Default = SNPA.

***Remote SNPA**

The next routing address (remote SNPA) of the next system to be accessed according to the SNPA type, Network type and Connection type fields.

MASK: A hexadecimal character chain including only the 0, f, F hexadecimal characters (the f, F must be consecutive). The maximum number of characters is equal to the length of the NSAP name. The address of the next system to be reached is retrieved from the NSAP name using this mask.

NET: A hexadecimal chain using the same syntax and meaning as the NSAP name (even number, minimum 2 characters, maximum 40).

SNPA – X.25 or LAN address: The address of the next system to be reached, using a hexadecimal character chain of:

12 characters for a Token Ring or an Ethernet address.
8 characters maximum for an X25 PVC name
15 characters for X.25 SVC.

Default = None.

***Subdomain Number**

A hexadecimal value from a ring showing the outgoing logical subdomain identifier, linking the local system to the next system to be reached. A list provides the defined subdomains with the following format:
<subdomain number> <adapter code> <adapter type>.

Default = 0.

***NSAP priority**

Allows the setting of NSAP routing priority, indicating preference of several routing paths. A decimal value in the range from 1 (highest priority) to 65 535.

Default = 1.

***RIB part**	A ring allows selection of the static mode: STATICV STATICF **STATICV**: Entry is part of the “variable” RIB on disk loaded into memory only on express request of the Inter–Network Protocol (CLNP). **STATICF**: Entry is part of the “fixed” RIB on disk and is automatically loaded into memory. Default = STATICF. It is recommended to use STATICF.
***State**	A ring allows selection of the current NSAP state in the RIB disk base: ENABLE LOCKED **ENABLE**: Authorizes the use of this routing path (default value). **LOCKED**: Prohibits the use of this routing path for administration purposes. Default = Enable.
***Message Lifetime**	Allows setting of the validity period of the elementary PDU transferred to the ISO 8473 protocol (CLNP). It determines the time–out before eliminating the packets lost because of non–efficient routing, or whose reference could be confused with more recent data. This period is measured by the number of passages in the Intermediate– and End–Systems. A decimal value in the range from 1 to 65535. Default value is 15.
CUG	On X.25 networks only. This parameter specifies the number of the Closed User Group (CUG) accepted on the connection. This information is inserted in the *facilities* field of the call packets used to set up the virtual circuit. A decimal number in the range 0 to 9999. Default value is 0.

Note: If an error occurs in the acquisition of a field, an error message is displayed. The user has to rectify the definition of the RIB entry. Otherwise a RIB entry is added.

Successful Result

The configuration name is displayed, followed by a confirmation message.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf
The RIB entry has been defined
```

How to Change/Show a Static RIB Entry

Access

From the RIB entries Management menu, select:

Change/show a Static RIB entry

FastPath: `ConfOSIAdrChR`

Command: `osirib` on page 10-26.

Overview

This command allows the RIB to be changed or displayed.

A selector provides a list of defined RIB entries. Each entry supplies the following information:

```
# Configuration name:   <conf_name>
# ----------------------------------------------------------------

<Internal Identifier>   # Internal ident
# NSAP: <NSAP name>
# Mask: <NSAP mask
# SNPA: <remote SNPA>
# Subdomain:    <Subdomain number>      Type:   <Network type>

<Internal Identifier>    .......
```

The internal identifier, a decimal number, is the one of the associated RIB entries on which the NSAP is defined.

If the configuration is the loaded one, the memory is updated.

Dialog Fields

See How to Add a Static RIB Entry on page 9-66.

Default values of the fields are those of the RIB entries already configured,

A RIB entry is modified if the value of a field is changed.

Successful Result

The configuration name is displayed, followed by a confirmation message.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

The RIB entry has been changed
```

How to Remove Static RIB Entries

Access

From the RIB entries Management menu, select:

Remove Static RIB entries

FastPath: `ConfOSIAdrRemR`

Command: `osirib` on page 10-26.

Overview

This command allows RIB entries to be deleted.

To delete a RIB entry, select one or more NSAPs that have a RIB entry defined from the list and confirm the selection.

If the configuration is the loaded one, the RIB in memory is updated.

Ghost Dialog Field

A selector provides a list of defined RIB entries. Each entry supplies the following information:

```
# Configuration name:   <conf_name>
# ---------------------------------------------------------------

<Internal Identifier>   # Internal ident
# NSAP: <NSAP name>
# Mask: <NSAP mask
# SNPA: <remote SNPA>
# Subdomain:    <Subdomain number>      Type:   <Network type>

<Internal Identifier>    .......
```

The internal identifier, a decimal number, is the one of the associated RIB entry on which the NSAP is defined.

Successful Result

The configuration name is displayed, followed by a confirmation message.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

The RIB entry of internal identifier 1 has been removed
```

Accessing Relay Entries Management

Access

From the Addressing Management menu, select:

Relay Entries Management

FastPath: `ConfOSIAdrRelays`

Overview

This menu allows the system administrator to manage Relay entries. Menu options are shown below:

Relay Entries Management
List defined Relay entries Add a Relay entry from a model Remove Relay entries

Description

The menu options are described below:

List defined Relay entries

Lists the defined relay entries on page 9-72.

Add a Relay entry from a model

Defines a relay route from a model on page 9-73.

Remove Relay entries

Removes defined relay entries on page 9-76.

How to List Defined Relay Entries

Access

From the Relay Entries Management menu, select:

List Defined Relay Entries

FastPath: `ConfOSIRelayList`

Command: `osirelay` on page 10-20.

Overview

This command consists in a ghost dialog.

Description

The list of the defined Relay entries with all attributes is displayed on the output, presented in creation order (the order in which objects have been added in the database).

Successful Result

The list of the defined Relay entries, with all attributes, is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

# List of the defined Relays
#-----------------------------------------------------------
RelayName          = <RelayName>
InTselEd           = <InTselEd>
InTselIng          = <InTselIng>
InAddrType         = <InAddrType>
InAddrIng          = <InAddrIng>
InNegoClass        = <InNegoClass>
OutTselEd          = <OutTselEd>
OutTselIng         = <OutTselIng>
OutAddrType        = <OutAddrType>
OutAddrEd          = <OutAddrEd>
OutAddrIng         = <OutAddrIng>
OutLsapEd          = <OutLsapEd>
   .......
```

How to Add a Relay Entry From a Model

Access

From the Relay Entries Management menu, select:

Add a Relay Entry From a Model

FastPath: `ConfOSIRelayAdd`

Command: `osirelay` on page 10-20.

Overview

This menu allows the definition of the adapter to be configured as one of the communications adapter types.

This menu begins with a selector which gives the list of the already defined Relay entries, where the user will select one to be used as a model. If the list is empty (first time the creation of a Relay entry is made), the command directly gives the dialog with default values for each field.

If the selected configuration is loaded, this Relay entry will be added in memory.

The list proposes all Relay name entries followed by the unique key made of the **InTselEd**, **InTselIng**, **InAddrType** and **InAddrIng** attributes. The Relay entries are presented in creation order (the order in which objects have been added in the database).

```
Configuration name  :  conf

List format for each Relay entry defined:
# Configuration name:   <conf-name>
# List of the defined Relay entries
# -------------------------------------------------------------
<RelayName>
#  InTselEd   = <InTselEd>        InTselIng    = <InTselIng>
#  InAddrType = <InAddrType>      InAddrIng    = <InAddrIng>

<RelayName>
   ...............
```

Then, there is a dialog with several fields.

Dialog Fields

The values displayed are those of the Relay entry selected as a model, or the default ones.

*** RelayName (Name of the Relay)**
A unique name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 10 characters).

InTselEd (Incoming called TSEL)
A hexadecimal string of up to 64 digits, or "0", or "–". (default "–").

InTselIng (Incoming calling TSEL)
A hexadecimal string of up to 64 digits, or "0", or "–". (default "–").

InAddrType (Incoming address type)
A ring gives the choice between:
– **NSAP**
– **LAN**
– **SVC**
– **PVC.**

InAddrIng (Incoming address)
if InAddrType is **NSAP**: hexadecimal string up to 40 digits (even number).
if InAddrType is **LAN**: hexadecimal string of 12 digits.
if InAddrType is **SVC**: decimal string up to 15 digits.
if InAddrType is **PVC**: A name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 8 characters).

InNegoClass (Incoming negotiation transport class)
A ring gives the choice between:
– **2**
– **3**
– **4** (default value).

OutTselIng (Outgoing calling TSEL)
A hexadecimal string up to 64 digits, or "0", or "–". Default "–".

*** OutAddrType (Outgoing address type)**
A ring gives the choice between:
– **SVC**
– **LANC**
– **CL**
– **LAN**
– **CLCO**
– **PVC.**

*** OutAddrEd (Outgoing called address)**
if OutAddrType is **CL** or **CLCO**: hexadecimal string up to 40 digits (even number).
if OutAddrType is **LANC** or **LAN**: hexadecimal string of 12 digits.
if OutAddrType is **SVC**: decimal string up to 15 digits.
if OutAddrType is **PVC**: A name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 8 characters).

OutAddrIng (Outgoing calling address or Subnet)
if OutAddrType is **CL** or **CLCO**: hexadecimal string up to 40 digits (even number) or the character "–".
if OutAddrType is **SVC**, **LANC** or **LAN**: A list proposes a

choice between the defined Subnetwork entries.
if OutAddrType is **PVC**: this parameter is not significant.

OutLsapEd (Outgoing LSAP used)

if OutAddrType is **CL or CLCO**: hexadecimal string of 2 digits or the character "–".
if OutAddrType is **PVC**, **LANC** or **LAN**: this parameter is not significant.
Default "–".

Successful Result

A Relay entry is added and a message to confirm the success of the task is displayed on the output.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Remove Relay Entries

Access

From the Relay Entries Management menu, select:

Remove Relay Entries

FastPath: `ConfOSIRelayRem`

Command: `osirelay` on page 10-20.

Overview

This menu consists of a selector where the Relay entries to be removed are selected.

Then, there is a dialog with several fields.

Dialog Fields

The values displayed are those of the Relay entry selected as a model, or the default ones.

A list proposes all Relay name entries followed by the unique key made of the **InTselEd**, **InTselIng**, **InAddrType** and **InAddrIng** attributes. The Relay entries are presented in creation order (the order in which objects have been added in the Data Base).

List format for each Relay entry defined:

```
# Configuration name:   <conf-name>
# List of the defined Relay entries
# ------------------------------------------------------------
<RelayName>#  InTselEd   = <InTselEd>            InTselIng    =
<InTselIng>#  InAddrType   = <InAddrType>      InAddrIng    =
<InAddrIng><RelayName>
    ........
```

If the configuration is loaded, the memory is updated.

Successful Result

Selected Relay entries are deleted and a message to confirm the success of the task is displayed on the output.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

Accessing Profile Entries Management

Access

From the Addressing Management menu, select:

Profile entries Management

FastPath: `ConfOSIAdrProfiles`

Overview

This menu allows the system administrator to manage profile entries, as shown below:

```
Profile entries Management
-----------------------------------------
List defined Profile entries
Add a Profile entry from a model
Remove Profile entries
```

Description

The menu options are described below:

List defined Profile entries

Lists defined Profile entries on page 9-78.

Add a Profile entry from a model

Allows a Profile entry to be added on page 9-79.

Remove Profile entries

Allows one or more profile entries to be removed on page 9-81.

How to List Defined Profile Entries

Access

From the Profile entries Management menu, select:

List defined Profile entries

FastPath: `ConfOSIProfList`

Command: `osiroute` on page 10-28.

Overview

This command lists the Profile entries with all attributes, in creation order (the order which objects have been added to the database).

Description

The existing subnets are displayed in a two–column format:

```
 Configuration name: <conf-name>
# List of the defined Profiles :
#-------------------------------------------------------
ProfileName       = <ProfileName>
Tclass            = <Tclass>
Taltclass         = <Taltclass>
UseExpData        = <UseExpData>
UseFlowCtrl       = <UseFlowCtrl>
UseExtendFmt      = <UseExtendFmt>
UseChecksum       = <UseChecksum>
UseAltRetranAlgo  = <UseAltRetranAlgo>
Priority          = <Priority>
Credit            = <Credit>
TPDUsize          = <TPDUsize>
CUD               = <CUD>
CUG               = <CUG>
   .......
```

How to Add a Profile Entry From a Model

Access

From the Profile entries Management menu, select:

Add a Profile entry from model

FastPath: `ConfOSIProfAdd`

Command: `osiroute` on page 10-28.

Overview

This menu allows an addition to be made to the Profile entries list.

This menu begins with a selector which gives the list of the already defined Profile entries, where the user will select one to be used as a model. If the list is empty (first time the creation of a Profile entry is made), the command gives directly the dialog with default values for each field.

If the selected configuration is loaded, this Profile entry will be added in memory.

The list proposes all Profile name entries with the following attributes: **Tclass**, **Taltclass**, **Priority**, **Credit**, **AltRetryAlgo** and **TPDUsize**.The Profile entries are presented in creation order (the order in which objects have been added in the Data Base).

```
Configuration name:   <conf-name>

List of the defined Profile entries
# -------------------------------------------------------------
<ProfileName>
# Tclass = <Tclass>  Taltclass = <Taltclass>  Priority =
<Priority>
#  Credit   = <Credit>      TPDUsize  = <TPDUsize>
AltRetryAlgo = <UseAltRetryAlgo>

<ProfileName>
      ...............
```

Then, there is a dialog with several fields the user will enter:

Dialog Fields

The values displayed are those of the Profile entry selected as a model, or the default ones.

***ProfileName (Name of the Profile)**
A unique name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 10 characters).

***Tclass (Transport class)**
A ring gives the choice between **2**, **3** and **4** (default value is **4**)

Taltclass (Alternative Transport class)
A ring gives the choice between **2**, **3**, **4** and **–1** (default value is **–1**)

UseExpData (Use expedited data ?)
A ring gives the choice between **Yes** and **No** (default value depends on the initial **Tclass** value)

UseFlowCtrl (Use Flow control ?)
A ring gives the choice between **Yes** and **No** (default value depends on the initial **Tclass** value)

UseExtendFmt (Use Extended control ?)
A ring gives the choice between **Yes** and **No** (default value depends on the initial **Tclass** value)

UseChecksum (Use Checksum ?)	A ring gives the choice between **Yes** and **No** (default value depends on the initial **Tclass** value)
UseAltRetranAlgo	A ring gives the choice between **Yes** and **No** (default value is **No**)
Priority	A ring gives the choice between **DFLT**, **TOP**, **HIGH**, **MID** and **LOW** (default value is **DFLT**)
Credit	A decimal value in the range 1 to 15. This parameter is not significant ifUseFlowCtrl is 0. (default value is 15).
TPDUsize	A ring gives the choice between the following values **128**, **256**, **512**, **1024**, **2048**, **4096** and **8192** (default value is: if initial **Tclass** value is 0, **2048**, for all other **Tclass** values, **8192**)
CUD	An hexadecimal string up to 8 digits. If the number of digits is odd, a 0 digit is added at the beginning of the string. (This parameter is only significant for an X25 Network)
CUG	A decimal value in the range of **–1** to **9999.** This parameter is only significant for an X25 Network. The value –1 means no CUG facility.

Successful Result

A Profile entry is added and a message to confirm the success of the task is displayed on the output.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Remove Profile Entries

Access

From the Profile entries Management menu, select:

Remove Profile entries

FastPath: `ConfOSIProfRem`

Command: `osiroute` on page 10-28.

Overview

This menus allows one or more Profile entries to be deleted.

If the configuration is loaded, the memory is updated.

The list proposes all Profile name entries with the following attributes:Tclass, Taltclass, Priority, Credit, AltRetranAlgo and TPDUsize.

The Profile entries are presented in creation order (the order in which objects have been added in the database).

List format for each Profile entry defined:

```
# Configuration name:   <conf-name>
# List of the defined Profile entries
# --------------------------------------------------------------
<ProfileName>
#  Tclass   = <Tclass>  Taltclass    = <Taltclass>
Priority      = <Priority>
#  Credit   = <Credit>   TPDUsize  = <TPDUsize>  AltRetranAlgo  =
<UseAltRetranAlgo>

<ProfileName>
     ........
```

Successful Result

The Profile entry(ies) is(are) deleted and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

Accessing Route Entries Management

Access

From the Addressing Management menu, select:

Route Entries Management

FastPath: `ConfOSIAdrRoutes`

Overview

This menu allows the system administrator to manage Route entries, as shown below:

```
Route Entries Management
---------------------------------
List defined Route entries
Add a Route entry
Remove Route entries
```

Description

The menu options are described below:

List defined Route entries

Lists the defined route entries on page 9-83.

Add a Route entry

Defines a route entry on page 9-84.

Remove Route entries

Removes defined route entries on page 9-85.

How to List Defined Route Entries

Access

From the Route entries Management menu, select:

List defined Route entries

FastPath: `ConfOSIRouteList`

Command: `osiroute` on page 10-28.

Overview

This command lists the defined Route entries with all attributes in creation order (the order in which objects have been added to the database).

Description

The existing Route entries are displayed in a two–column format:

```
 Configuration name: <conf-name>
# List of the defined Routes:
#-------------------------------------------------------
RouteName          = <RouteName>
InternProfile      = <InternProfile>
RouteType          = <RouteType>
RouteValue         = <RouteValue>
   .......
```

How to Add a Route Entry

Access

From the Route entries Management menu, select:

Add a Route entry

FastPath: `ConfOSIRouteAdd`

Command: `osiroute` on page 10-28.

Overview

This menu allows an addition to be made to the Route entries list.

This menus begins with a selector which gives the list of the already defined Profile entries, where the user will select one on which the Route is going to be defined. If the list is empty, no Routes can be defined.

The list proposes all Profile name entries with the following attributes: **Tclass**, **Taltclass**, **Priority**, **AltRetranAlgo**, **Credit** and **TPDUsize**.The Profile entries are presented in creation order (the order in which objects have been added in the Data Base).

List format for each Profile entry defined:

```
Configuration name:   <conf-name>

List of the defined Profile entries
# ------------------------------------------------------------<
ProfileName>
#  Tclass   = <Tclass>  Taltclass    = <Taltclass>
Priority   = <Priority>
#  Credit   = <Credit>  TPDUsize  = <TPDUsize>
AltRetranAlgo  = <UseAltRetranAlgo>

<ProfileName>
      ...............
```

Then, there is a dialog with several fields the user will enter:

Dialog Fields

The values displayed are the default values.

***RouteName (Name of the Route)**	A unique name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 10 characters).
ProfileName	Only for information
***RouteType (Type of the Route)**	A ring gives the choice between **NSAP**, **LAN**, **SVC** and **PVC**
***RouteValue (Value of the Route)**	according to the value of **RouteType** if **NSAP**, hexadecimal string up to 40 digits (even number). if **LAN**, hexadecimal string of 12 digits. if **SVC**, decimal string up to 15 digits. if **PVC**, a name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 8 characters).

Successful Result

A Route entry is added and a message to confirm the success of the task is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Remove Route Entries

Access

From the Route entries Management menu, select:

Remove Route entries

FastPath: `ConfOSIRouteRem`

Command: `osiroute` on page 10-28.

Overview

This command allows one or more route entries to be deleted.

If the configuration is loaded, the memory is updated.

The list proposes all Route name entries followed by the RouteType and RouteValue attributes. The Route entries are presented in creation order (the order in which objects have been added in the Data Base).

List format for each Route entry defined:

```
# Configuration name:   <conf-name>
# List of the defined Route entries
# -------------------------------------------------------------
<RouteName>
#  RouteType  = <RouteType>       RouteValue  = <RouteValue>

<RouteName>
    ........
```

Successful Result

The Route entry(ies) is(are) deleted and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

Accessing Security Filters Management

Access
From the Addressing Management menu, select:

Security Filter Management

FastPath: `ConfOSISecurity`

Overview
This menu allows the system administrator to manage Security Filters. Menu options are shown below:

Security Filter Management
List defined Filters entries Add a Filter entry from a model Change/Show Security on layers Remove Filters entries

Description

The menu options are described below:

List defined Filters entries

Lists the subnet table in creation order (the order which objects have been added to the database) on page 9-87.

Add a Filter entry from a model

Allows a filter entry to be added on page 9-88.

Change/Show Security on layers

Allows the filter declared in the database (associated with an OSI Configuration) to be shown and taken into account on page 9-90.

Remove Filters entries

Allows one or more filter entries to be removed on page 9-91.

How to List Defined Security Filter Entries

Access

From the Security Filter Management menu, select:

List defined Filter entries

FastPath: `ConfOSIFilterList`

Command: `osifilter` on page 10-12.

Overview

This command consists in a ghost dialog. The list of the Filter entries, with all attributes, is displayed on the output presented in creation order (the order in which objects have been added in the database).

Description

The Filter entries are displayed in a two–column format:

```
 Configuration name: <conf-name>
# List of the defined Filters :
#-------------------------------------------------------
Filter Name          = <FilterName>
Called Tsel          = <CalledTsel>
Addresses Type       = <AddrType>
Called Address       = <CalledAddr>
Calling Address      = <CallingAddr>
   .......
```

How to Add a Security Filter Entry From a Model

Access

From the Subnetwork entries Management menu, select:

Add a Filter entry

FastPath: `ConfOSIFilterAdd`

Command: `osifilter` on page 10-12.

Overview

This menu allows a filter entry to be added.

This command begins with a selector which gives the list of the already defined Filter entries, where the user selects one to be used as a model.

If the list is empty (first time the creation of a Filter entry is made), the command directly gives the dialog with default values for each field.

If the selected configuration is loaded, this Filter entry will be added in memory.

Note: The NULL TSEL is defined with the string ”0” (zero), and all wildcard values with the string ”–”.

The list proposes all Filter name entries followed by the unique key made of the **AddrType**, **CallingAddr**, **CalledTsel** and **CalledAddr** attributes.

The Filter entries are presented in creation order (the order in which objects have been added in the database).

List format for each Filter entry defined:

```
# Configuration name:   <conf-name>
# List of the defined Filter entries
# -------------------------------------------------------------
<FilterName>
#  Called Tsel       = <CalledTsel>
#  Addresses Type    = <AddrType>
#  Called Address    = <CalledAddr>
#  Calling Address   = <CallingAddr>

<FilterName>
      ...............
```

Then, there is a dialog with several fields.

Dialog Fields

The values displayed are those of the Filter entry selected as a model, or the default ones.

*** Filter Name**	A unique name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 10 characters).
*** Called TSEL)**	A hexadecimal string of up to 64 digits, or ”0”, or ”–”. Default ”–”.
Address Type	A ring gives the choice between: – **NSAP** – **LAN** – **SVC** – **PVC.**

Called Address	if InAddrType is **NSAP**: hexadecimal string up to 40 digits (even number). if InAddrType is **LAN**: hexadecimal string of 12 digits. if InAddrType is **SVC**: decimal string up to 15 digits. if InAddrType is **PVC**: A name in an alphanumeric character string including '_' '.' '/' and '\' (1 to 8 characters). Default "–".
Calling Address	if InAddrType is **NSAP**: hexadecimal string up to 40 digits (even number). if InAddrType is **LAN**: hexadecimal string of 12 digits. if InAddrType is **SVC**: decimal string up to 15 digits. if InAddrType is **PVC**: This field is irrelevant. Default "–".

Successful Result

A filter entry is added and a message to confirm the success of the task is displayed on the output.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Change/Show Security Filters on Layers

Access

From the Security Filter Management menu, select:

Change/Show Security Filters on Layers

FastPath: `ConfOSIFilterState`

Command: `osifilter` on page 10-12.

Overview

This command consists in a ghost dialog which allows the filter declared in the database (associated with an OSI Configuration) to be shown and taken into account.

This command allowed to enable or disable the filtering on the layers COTP, CLTP or CLNP.

The values displayed are the current values. The first times the values are OFF.

Dialog Fields

***Filters on COTP connection establishment**

Select from ring:

ON

OFF.

Default value = OFF.

***Filters on CLTP received TSDU**

Select from ring:

ON

OFF.

Default value = OFF.

***Filters on CLNP Routed NPDU**

Select from ring:

ON

OFF.

Default value = OFF.

Successful Result

A message to confirm the success of the task is displayed on the output. In case of failure an explanation is given.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Remove Security Filter Entries

Access

From the Security Filter Management menu, select:

Remove Filter entries

FastPath: `ConfOSIFilterRem`

Command: `osifilter` on page 10-12.

Overview

This command allows one or more filter entries to be deleted.

If the configuration is loaded, the memory is updated.

The list proposes all Filter name entries followed by the unique key made of the CallingAddr, CalledTsel and CalledAddr attributes.

The Filter entries are presented in creation order (the order in which objects have been added in the database).

List format for each Filter entry defined:

```
# Configuration name:   <conf-name>
# List of the defined Filter entries
# --------------------------------------------------------------
<FilterName>
#  Called Tsel          = <CalledTsel>
#  Addresses Type   = <CallingAddrType>
#  Called Address    = <CalledAddr>
#  Calling Address  = <CallingAddr>

<FilterName>
    ........
```

Successful Result

Filter entries are deleted and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Configure OSI Layers

Access

From the Configuration definition menu, select:

OSI Layers Configuration

FastPath: `ConfOSILayers`

Command: `osiconf` on page 10-4.

Overview

It is from this menu that the basic configuration parameters of the OSI Stack layers are accessed.

Dialog Fields

Max. no. of simultaneous COPP connections

Enter a decimal value in the range from 1 to 1024. Default value is 64.
This value cannot be greater than the maximum number of simultaneous COSP connections.
This parameter is updated in memory if the selected configuration is the loaded one.

Max. no. of simultaneous COSP connections

Enter a decimal value in the range from 1 to 1024. Default value is 64.
This parameter is updated in memory if the selected configuration is the loaded one.

Max. no. of simultaneous COTP connections

Enter a decimal value in the range from 1 to 1024. Default value is 64.
This parameter is updated in memory if the selected configuration is the loaded one.

Maximum number of OSI COTP connections multiplexed on a VC

Enter a decimal value in the range from 1 to 1024. Default value is 64.
This parameter cannot be modified in memory. If the configuration is loaded, the command is rejected.

Use of the ES IS Protocol

Select from ring:
No
Yes.
Default value is **Yes**.
Allows the ES–IS protocol to be activated or de–activated.
This parameter is updated in memory if the selected configuration is the loaded one.

Type of function assumed on system	Type of function assumed by the local machine in the OSI network configuration. Select from ring: **ES** **IS** **ES+IS.** Where: ES = COTP and upper OSI layers services only. IS = Routing Exchange service only. ES+IS = Both COTP with upper OSI layers services and Routing Exchange service (full OSI service). Default value is **ES**. This parameter cannot be modified in memory. If the configuration is loaded, the command is rejected.

The default values are those of the basic configuration or those already contained in the configuration. Displayed default values are those previously configured.

Successful Result

The name of the updated configuration is displayed, followed by a confirmation message.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

The selected configuration has been modified.
```

Accessing X.25 Configuration

Access

From the Configuration definition menu, select:

X25 Configuration

FastPath: `ConfOSIX25`

Overview

It is from this menu that additional X.25 Network parameters, not supplied with the machine are configured. The commands held in the menus affect only the database. The X.25 Configuration menu is shown below:

X.25 Configuration
Permanent Virtual Circuit (PVC)Management Subsequent Protocol Identifier (SPI) Table

Description

If the configuration is loaded, only the List commands are available. The menu options are described below:

Permanent Virtual Circuit (PVC) Management

The PVC table associates names with each subscription see page 9-95.

Subsequent Protocol Identifier (SPI) Table

The SPI table ensures the routing of incoming calls see page 9-100.

Note: A configuration must already be selected, see How to Select a Configuration on page 9-44.

Accessing Permanent Virtual Circuit (PVC) Management

Access

From the X25 Configuration menu, select:

Permanent Virtual Circuit (PVC) Management

FastPath: `ConfOSIPVC`

Overview

It is through this menu that PVC parameters are managed, The PVC Management menu is shown below:

Permanent Virtual Circuit (PVC) Management
List the PVCs Add a PVC Change / Show a PVC Remove a PVC

Description

If the configuration is loaded, only the List commands are available. The menu options are described below:

List the PVCs
Lists the PVCs see page 9-96.

Add a PVC
Adds a PVC see page 9-97.

Change / Show a PVC
Displays or modifies a PVC see page 9-98.

Remove a PVC
Removes a PVC see page 9-99.

How to List PVCs

Access

From the Permanent Virtual Circuit (PVC) Management menu, select:

List the PVCs

FastPath: `ConfOSIPVCL1`

Command: `osix25 -l <adaptercode>`
on page 10-36.

Overview

Select the X25 adapter code from the list.

Dialog Fields

*** Adapter code**	A list proposes the X25 communications adapter already configured in the configuration. Format is as follows: <adapter code> <adapter type>.

Successful Result

The list of the defined PVCs in the selected communications adapter is displayed on the output.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
  Configuration name  :  conf

PVC-NAME    ADAPTERCODE  CHANNEL     OWNER
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

toto        00-02-01-00    1         3610
```

How to Add a PVC

Access

From the Permanent Virtual Circuit (PVC) Management menu, select:

Add a PVC

FastPath: `ConfOSIPVCA1`

Command: `osix25` on page 10-36.

Overview

Select the X25 adapter code from the list. The maximum number of PVC configured depends on the communications adapter and device driver used.

Refer to the specifications for X25 communications adapters.

If the configuration is already loaded, the command will be rejected.

Dialog Fields

*** Adapter code**	For information only. A list proposes the X25 communications adapter already configured in the configuration. Format is as follows: <adapter code> <adapter type>.
*** PVC Name**	Name of the PVC in a string of 1 to 8 alphanumeric characters, including only '_' '.' '/' and '\'.
*** Logical Channel Number**	Logical Channel Number (LCN), a decimal value in the range from 0 to 4095. **Note:** The Channel Number for the HiSpeed WAN Comm. Adapter must be in the range from **1 to 4095**.
*** Module Owner**	A list proposes the choice: – 0000: Not used – 0410: COTP – 3610: Access Method X25.3.

Successful Result

A PVC is added to the Subscription.
A message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Change/Show a PVC

Access

From the Permanent Virtual Circuit (PVC) Management menu, select:

Change/Show a PVC

FastPath: `ConfOSIPVCC1`

Command: `osix25 -s <adaptercode> <PVCname> <channel> <owner>` on page 10-36.

Overview

Select the X25 adapter code from the list. Then select the PVC to be modified or shown. A list proposes the PVC already configured in the selected X.25 communications adapter.

```
#configuration name: <conf-name>
#PVC-NAME   ADAPTERCODE     CHANNEL    OWNER
#-----------------------------------------------------
-<pvc-name> <adapter-code>  <channel>  <owner>
```

If the configuration is already loaded, the command will be rejected.

Dialog Fields

*** Adapter code**	For information only. Shows the X25 adapter previously selected in list. Format is as follows: <adapter code> <adapter type>.
*** PVC Name**	For information only. Name of the selected PVC.
*** Logical Channel Number**	Logical Channel Number (LCN), a decimal value in the range of 0 to 4095. **Note:** The Channel Number for the HiSpeed WAN Comm. Adapter must be in the range from **1 to 4095**.
*** Module Owner**	A list proposes the choice: – 0000: Not used – 0410: COTP – 3610: Access Method X25.3.

Successful Result

The selected PVC is modified and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Remove a PVC

Access

From the Permanent Virtual Circuit (PVC) Management menu, select:

Remove a PVC

FastPath: `ConfOSIPVCR1`

Command: `osix25` on page 10-36.

Overview

Select X25 adapter codes from the list. Then select the PVC to be removed. A list proposes the PVC already configured in the selected X.25 communications adapter.

```
configuration name: <conf-name>
PVC-NAME   ADAPTERCODE      CHANNEL    OWNER
------------------------------------------------------
<pvc-name> <adapter-code>  <channel>  <owner>
```

Select one or more PVCs from the list. Confirm selection.

If the configuration is already loaded, the command will be rejected.

Successful Result

The selected PVCs are removed from the Subscription and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Set Up the Subsequent Protocol Identifier (SPI) Table

Access

From the X.25 configuration menu, select:

Subsequent Protocol Identifier (SPI) Table

FastPath: `ConfOSISPI`

Command: `osiconf` on page 10-4.

Overview

The SPI table defines the routing of incoming calls. The SPI value is the prefix of an address data field that determines which incoming call packets are authorized.

For each SPI value, the associated module where packets are routed, is specified. Two entries describe where to route packets that have no data field and packets that have an unknown SPI.

A configuration must already be selected, see How to Select a Configuration on page 9-44.

If the configuration is already loaded, the command will be rejected.

Dialog Fields

***Module name if there is no DATA field in a packet**

A list proposes the following values:
0000 : Not used
0410 : COTP
0F10 : ISO 8473 (CLNP)
3610 : Access Method X25.3.

The default value is hex 0410.

***Module when unknown SPI**

A list proposes the following values:
0000 : not used
0410 : COTP
0F10 : ISO 8473 (CLNP)
3610 : X.25.3 Access Method

The default value is hex 3610.

***SPI 1 value**

Enter a hexadecimal value in 8 digits max.

The default value is hex 0.

***Module associated to the SPI 1**

A list proposes the following values:
0000 : Not used
0410 : COTP
0F10 : ISO 8473 (CLNP)
3610 : Access Method X25.3.

The default value is hex 0410.

***SPI 16 value**

Enter a hexadecimal value in 8 digits max.

The default value is hex 0.

***Module associated to the SPI 16**

A list proposes the following values:
0000 : Not used

0410 : COTP
0F10 : ISO 8473 (CLNP)
3610 : Access Method X25.3.

The default value is hex 3610.

The default values are those of the basic configuration, defined by the OSI standards. The displayed default values are those already contained in the configuration.

SPI/Hexadecimal/Module Equivalence

SPI Number	SPI Value (Hex)	Module associated with the SPI value (Hex)
no DATA	–	
unknown SPI	–	0410 COTP
1	00	3610 X25.3 Access Method
2	01	0410 COTP
3	02	3610 X25.3 Access Method
4	03	3610 X25.3 Access Method
5	81	0410 COTP
6	01	0F10 CLNP
7	01	3610 X25.3 Access Method
8	01	3610 X25.3 Access Method
9	01	3610 X25.3 Access Method
10	01	3610 X25.3 Access Method
11	01	3610 X25.3 Access Method
12	01	3610 X25.3 Access Method
13	01	3610 X25.3 Access Method
14	01	3610 X25.3 Access Method
15	01	3610 X25.3 Access Method
16	01	3610 X25.3 Access Method
		3610 X25.3 Access method

Note: It is usually unnecessary to change the default SPI definitions.

Successful Result
A message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name  :  conf

The Subsequent Protocol Identifier Table has been modified.
```

Accessing Configuration Tuning

Access

From the Configuration Definition menu, select:

Tuning Configuration

FastPath: `ConfOSITune`

Overview

It is from this menu configuration tuning parameters can be defined.

CAUTION

Configuration tuning is reserved for experienced Administrators. Modifying timers or other parameters without complete understanding of their roles may lead to malfunction of Stack services.

The Tuning Configuration menu is shown below:

```
Tuning configuration
---------------------------------------------------
Connection Oriented Session Protocol (COSP)
Connection Oriented Transport Protocol (COTP)
End System - Intermediate System (ES-IS)
Memory Management
Null CNLP Profile Network LSAPs definition
```

Description

The menu options are described below:

Connection Oriented Session Protocol (COSP)
COSP protocol see page 9-103.

Connection Oriented Transport Protocol (COTP)
COTP protocol see page 9-104.

Memory Management
Memory management see page 9-106.

Null CLNP Profile Network LSAPs definition
Null CLNP profile network LSAPs see page 9-107.

How to Tune Connection Oriented Session Protocol (COSP)

Access

From the Tuning Configuration menu select:

Connection Oriented Session Protocol

FastPath: `ConfOSITuneSess`

Command: `osiconf` on page 10-4.

Overview

The connection timer is the maximum time allowed to establish a connection.

The disconnection timer is the maximum time allowed to break a connection.

The indication connection timer is the maximum time allowed to store a connection indication when there is no recipient.

This command allows the timer interval to be modified.

Dialog Fields

#COSP connection timer
Enter a decimal value in the range 160 to 20 000 seconds.
Default value is 160 seconds.
If the configuration is loaded, this timer can be modified.

#COSP disconnection timer
Enter a decimal value in the range 10 to 20 000 seconds.
Default value is 40 seconds.
If the configuration is loaded, this timer can be modified.

#COSP indication connection lifetime
Enter a decimal value in the range 1 to 20 000 seconds.
Default value is 5 seconds.
CAUTION: If the configuration is loaded, this timer cannot be modified.

The default values are those of the basic configuration. The displayed default value is that previously configured.

Successful Result

The parameter values are modified and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Tune Connection Oriented Transport Protocol (COTP)

Access

From the Tuning Configuration menu select:

Connection Oriented Transport Protocol

FastPath: `ConfOSITuneTrans`

Command: `osiconf` on page 10-4.

Overview

The COTP timer measures the time interval before repeating the Protocol Data Unit (PDU).

This command allows the timer interval to be modified for both LAN and WAN.

This command also allows the maximum number of transmissions of a PDU to be modified (ISO/IEC Standard DIS8073 parameter N).

Dialog Fields

COTP timer (T1) on LAN
Enter a decimal value in the range 500 to 30 000 milliseconds.
Default value is 3 000 milliseconds.
This parameter is updated in memory if the selected configuration is the loaded one.

COTP timer (T1) on WAN
Enter a decimal value in the range 500 to 60 000 milliseconds.
Default value is 15 000 milliseconds.
This parameter is updated in memory if the selected configuration is the loaded one.

Max. no. of transmissions of a COTP LAN PDU (N)
Enter a decimal value in the range 1 to 15.
Default value is 10.
This parameter is updated in memory if the selected configuration is the loaded one.

The default values are those of the basic configuration. The displayed default value is that previously configured.

Successful Result

The parameters are modified in the memory if the configuration is loaded. A message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Tune ES–IS

Access

From the Tuning Configuration menu select:

End System – Intermediate System (ES–IS)

FastPath: `ConfOSITuneESIS`

Command: `osiconf` on page 10-4.

Overview

The ES timer measures the time interval between each PDU ESH.

The IS timer measures the time interval between each PDU ISH.

This command allows the timer interval to be modified.

If the configuration is the loaded one, the command will be rejected.

Dialog Fields

***ES timer**	Enter a decimal value in the range 1 to 20 000 seconds. Default value is 180 seconds.
***IS timer**	Enter a decimal value in the range 1 to 20 000 seconds. Default value is 180 seconds.

The default values are those of the basic configuration. The displayed default value is that previously configured.

Successful Result

The parameter values are modified and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Use Memory Management

Access

From the Tuning Configuration menu select:

Memory management

FastPath: `ConfOSITuneMemo`

Command: `osiconf` on page 10-4.

Overview

The buffer memory size represents the capacity of the data buffer where waiting frames, contexts and other data in the communication system are stored.
There is a default value for this memory, according to the number of connections. Only the lower layers use this buffer memory.
As the default value may be too big in particular cases, the user may choose another value. To help him in his choice, he can consult the Change/Show Loaded Configuration menu menu on page 9-113.

Dialog Fields

***Buffer memory size (Kb)**	Enter a decimal value in the range 1 000 to 1 000 000 Kbytes or the word ”automatic”. **Note:** With the word ”automatic”, the memory size is calculated automatically at loading time, or when modifying the loaded configuration. This is the default value of the basic configuration.

This parameter will be updated in memory if the selected configuration is the loaded one.

The default values are those of the basic configuration. The displayed default value is that previously configured.

Successful Result

The parameter value is modified and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Define NULL CLNP Profile Network LSAPs

Access

From the Tuning Configuration menu select:

Memory management

FastPath: `ConfOSITuneLSAP`

Command: `osiconf` on page 10-4.

Overview

These Local Service Access Points (LSAPs) are used to route the received PDU to the transport in a NULL CLNP profile network.

The command is rejected if the selected configuration is already loaded.

Dialog Fields

NULL CLNP profile network LSAPs
A list of 2 hexadecimal digits, up to 16 times, separated by a space. All couples of digits need to be different. Default value shown is that already defined, or the LSAP value 20.

Successful Result

Received PDU route is defined.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

Accessing Netbios Configuration

Access

From the Configuration Definition menu, select:

Netbios Configuration

FastPath: `ConfOSINetbios`

Overview

This menu allows the system administrator to manage the multicast address used by Netbios, as shown below:

```
Netbios Configuration
---------------------
Show Netbios configuration
Define Netbios configuration
```

Description

The menu options are described below:

Show Netbios configuration

Shows the defined Netbios attributes on page 9-109.

Define Netbios configuration

Lists the defined Subnet entries and permits definition of Netbios multicast configuration on page 9-110.

How to Show Netbios Configuration

Access

From the Netbios Configuration menu, select:

Show Netbios Configuration

FastPath : `ConfOSINbiosShow`

Command: `osinetbios` on page 10-16.

Overview

This command shows Netbios attributes.

```
Configuration name: <conf-name>
       Netbios Subnet number      : <0x...>
       Netbios NSAP               : <0x...>
       Netbios LSAP               : <..>
```

Successful Result

The Netbios multicast configuration is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Define Netbios Configuration

Access

From the Subnetwork entries Management menu, select:

Add a Subnetwork entry

FastPath : `ConfOSINbiosDef`

Command: `osinetbios` on page 10-16.

Overview

If the selected configuration is loaded or if the Subnet list is empty, the Netbios multicast configuration cannot be defined.

This menu begins with a selector which gives the list of the already defined Subnet entries, where the user selects one for Netbios.

Format:

```
# Configuration name:   <conf_name>
# SUBDOMAIN         LOCATION              ADAPTERTYPE
# ----------------------------------------------------------------
<subdomain number>  <adapter code>      <adapter type>
    ..........
```

Then, there is a dialog with several fields:

Dialog Fields

The values displayed are the default values.

Subdomain number	Only for information (The previously selected Subdomain number)
Netbios NSAP value	hexadecimal string up to 40 digits (even number). (Default value: the already defined value or the <49000006a00010001> value)
Netbios LSAP value	hexadecimal string of 2 digits. (Default value: the already defined value or the <ec> value)

Successful Result

Netbios multicast configuration is defined and a message to confirm the success of the task is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

Unloading Current Loaded Configuration

Access

From the OSI Configuration menu select:

Unload current loaded configuration

FastPath: `ConfOSIUnload`

Command: `osiunload` on page 10-35.

Overview

This command allows the OSI Stack to be de–activated.
All known running applications must be de–activated, otherwise a warning message recommends the user to de–activate them, and confirmation is asked from the user before the OSI Stack services are stopped.

Description

This command consists in a ghost dialog; confirmation is asked of the user before execution of the command.

All applications registered in the Start Applications Service (SAS) are automatically stopped before trying to unload the stack.

Successful Result

The OSI Stack is de–activated and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Command: OK      stout:  yes      stderr:  no

Before command completion, additional instructions may appear
below.

Configuration name: OSIdefault
OSI Stack unloaded.
```

Loading Last Selected Configuration

Access

From the OSI Configuration menu select:

Load last selected configuration

FastPath: `ConfOSILoad`

Command: `osiload` on page 10-15.

Overview

This command allows the OSI Stack, which has been previously defined and selected from a given configuration, to be activated.

Description

This command consists in a ghost dialog; confirmation is asked of the user before the OSI Stack is loaded and activated.

All applications registered in the Start Applications Service (SAS) are activated.

Successful Result

The OSI Stack is activated and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Command: OK      stout:  yes      stderr:  no

Before command completion, additional instructions may appear
below.

Configuration name: OSIdefault
OSI Stack loaded.
```

Accessing Change/Show Loaded Configuration

Access

From the OSI Configuration menu, select:

Change/Show Loaded configuration

FastPath: `ConfOSILoadedMag`

Overview

This screen provides the means to display dynamic RIB entries, reset static and dynamic RIB, show the state of the ES–IS protocol and show the memory buffer status. The Change/Show Loaded Configuration menu is shown below:

```
Change/Show Loaded configuration
```
```
List Waiting RIB Entries
List Dynamic RIB Entries
List Static RIB Entries
Show a Dynamic RIB Entry
Reset Dynamic RIB
Reset Static RIB
Show ES–IS Status
Show Memory buffer status
Change Subnet State
```

Description

The commands held in this menu affect only the Loaded Configuration (i.e. the memory).

The menu options are described below:

List Wait RIB Entries
Lists the RIB entries loaded in memory waiting to be resolved, see page 9-115.

List Dynamic RIB Entries
Lists Dynamic RIB entries which are gathered automatically when ES–IS protocol is enabled, see page 9-116.

List Static RIB Entries (in memory)
Lists Static RIB entries in memory, see page 9-64.

Show a Dynamic RIB Entry
Displays details of selected Dynamic RIB entry which is gathered automatically when ES–IS protocol is enabled, see page 9-119.

Reset Dynamic RIB
Removes Dynamic RIB entries from CPU memory, see page 9-120.

Reset Static RIB
Removes Static RIB entries from CPU memory, see page 9-121. The Static fixed base entries are automatically reloaded from the RIB on disk, see page 9-121.

Display ES–IS Status
Displays the state of the OSI protocol, see page 9-122.

Show Memory Buffer Status
Displays information about buffer memory size and occupied memory, see page 9-123.

Change Subnet State
Allows to change a subnet status of an adapter in memory, see page 0 .

How to List Waiting RIB Entries

Access

From the Change/Show Loaded Configuration menu, select:

List WAIT RIB entries

FastPath: `ConfOSIListWaitRIB`

Command: `osisnapshot` on page 10-33.

Overview

This command lists the RIB entries loaded in memory waiting to be resolved by the *osiribd* daemon process or by the ES–IS protocol. See Routing Framework and Routing Information Base, on page 3-10.
It consists of a ghost dialog.
The list of the Wait RIB entries is displayed with:

```
Configuration name:  <conf-name>

Internal Identifier     : <internal identifier>
NSAP Name               : <NSAP name>
State                   : <entry state: WAIT>
Type                    : <entry type: DYNAMIC, STATICV or
                           STATIF>
NSAP Holding Time       : <NSAP holding time entry in seconds if
                           the entry is a DYNAMIC RIB entry>
```

Successful Result

The command displays a list of the RIB entries waiting in memory.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name:  OSIdefault

Internal Identifier     : 3
NSAP Name               : 1111
State                   : WAIT
Type                    : DYNAMIC
NSAP Holding Time       : 180
```

How to List Dynamic RIB Entries

Access

From the Change/Show Loaded Configuration menu, select:

List Dynamic RIB entries

FastPath: `ConfOSIListDynRIB`

Command: `osisnapshot` on page 10-33.

Overview

This command lists the Dynamic RIB entries loaded in memory.
It consists of a ghost dialog.
The list of the Dynamic RIB entries is displayed with:

```
Configuration name:  <conf-name>

Internal Identifier     : <internal identifier>
NSAP Name               : <NSAP name>
Remote SNPA Type        : <type of remote SNPA: SNPA,  MASK or
                           NET>
Remote SNPA             : <address to next system to be reached
                          if the Remote SNPA type is SNPA,
                           otherwise field not displayed
Subdomain Number        :  <subdomain number>
System Type             : <type of the remote system: ES, IS,
                           ES-IS>
Usage Time              : <usage time of information in seconds>
NSAP Holding Time       : <NSAP holding time entry in seconds>
```

Successful Result

The command displays a list of the Dynamic RIB entries.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name:  test3

Internal Identifier     : 1
NSAP Name               : 0x12233445
Remote SNPA Type        : SNPA
Remote SNPA             : 0x123456789012
Subdomain Number        : 0x100
System Type             : IS
Usage Time              : 720
NSAP Holding Time       : 180
```

How to List Static RIB Entries (from memory)

Access

From the Change/Show Loaded Configuration menu, select:

List Static RIB entries

FastPath: `ConfOSIListStaRIB`

Command: `osisnapshot` on page 10-33.

Overview

This command lists the Static RIB entries loaded in memory.
It consists of a ghost dialog.
The list of the Static RIB entries is displayed with:

```
Configuration name:  <conf-name>

Internal Identifier     : <internal identifier>
NSAP Name               : <NSAP name>
NSAP Type               : <NSAP type
NSAP mask               : <mask value>
Network type            : <Network type>
Remote SNPA Type        : <type of remote SNPA: SNPA, MASK or
                           NET>
Remote SNPA             : <address to next system to be reached>
Subdomain number        : <Subdomain number>
System Type             : <type of the remote system: ES, IS,
                           ES-IS>
NSAP priority           : <priority>
RIB part                : <STATICV or STATICF>
State                   : <state>
Message lifetime        : <lifetime value>
CUG                     : <Close User Group value>
Usage Time              : <usage time of information in seconds>
NSAP Holding Time       : <NSAP holding time entry in seconds>
```

Successful Result

The command displays a list of the Static RIB entries (loaded in memory).

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name:  OSIdefault

Internal Identifier     : 1
NSAP Name               : 0x123456
NSAP Type               : NSAP
NSAP mask               : 0xFFFFFF
Network type            : ETHERNET
Remote SNPA Type        : SNPA
Remote SNPA             : 0x123456789012
Subdomain number        : 0x100
System Type             : ES
NSAP priority           : 5
RIB part                : STATICF
State                   : ENABLE
Message lifetime        : 15
CUG                     : 0
Usage Time              : 720
NSAP Holding Time       : 180
```

How to Show a Dynamic RIB Entry

Access

From the Change/Show Loaded Configuration menu, select:

Sho Dynamic RIB entry

FastPath: `ConfOSIShowDynRIB`

Command: `osisnapshot` on page 10-33.

Overview

This command shows the contents of the selected Dynamic RIB entry.
Select Dynamic RIB entry from list.

```
#Configuration name:  test3
#   Ident  Subdomain     NSAP       SNPA
      1       100     11223344   123456789012
```

The contents of the selected Dynamic RIB entries is displayed.

Successful Result

The command shows the contents of the selected Dynamic RIB.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name:  test3
Entry in Dynamic RIB
Internal Identifier     : 1
NSAP Name (ASCII)       : 9 ....ENE........8........
NSAP Name (NORMAL)      : 0x11223344
Remote SNPA             : 0x123456789012
NSAP type               : NSAP
Remote SNPA type        : SNPA
State                   : ENABLE
Subdomain Number        : 0x100
NSAP in use             : 720
NSAP Holding Time       : 300
Network type            : ETHERNET
Connection type         : SVC
System Type             : IS
```

How to Reset Dynamic RIB

Access

From the Change/Show Loaded Configuration menu, select:

Reset Dynamic RIB

FastPath: `ConfOSIResetDynRIB`

Command: `osisnapshot` on page 10-33.

Overview

This command deletes the Dynamic RIB entries in the kernel memory.

Successful Result

A message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Reset Static RIB

Access

From the Change/Show Loaded Configuration menu, select:

Reset Dynamic RIB

FastPath: `ConfOSIResetStaRIB`

Command: `osisnapshot` on page 10-33.

Overview

This command deletes the Static RIB entries in the kernel memory. The Static fixed base entries are automatically reloaded from the RIB on disk.

Successful Result

A message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

How to Display ES–IS Status

Access

From the Change/Show Loaded Configuration menu, select:

Show ES–IS Status

FastPath: `ConfOSIShowESIS`

Command: `osisnapshot` on page 10-33.

Overview

This command displays ES–IS status information.

Successful Result

Status information is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name: Confname
System type     : ES
Status          : active
```

How to Show Memory Buffer Status

Access

From the Change/Show Loaded Configuration menu, select:

Show Memory Buffer Status

FastPath: `ConfOSIShowMem`

Command: `osisnapshot` on page 10-33.

Overview

This command displays buffer memory size and occupied memory information.

```
Configuration name              : <conf-name>
Maximum memory                  : xxx Kbytes
Memory currently allocated      : xxx Kbytes
Memory currently used           : xxx Kbytes
```

Successful Result

Status information is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Configuration name              : confname
Maximum memory                  : 86016 Kbytes
Memory currently allocated      : 319 Kbytes
Memory currently used           : 37 Kbytes
```

How to Change a Subnet State

Access

From the Change/Show Loaded Configuration menu, select:

Change Subnet State

FastPath : `ConfOSIChnSubd`

Command: `osisnapshot` on page 10-33**.**

Overview

This command allows to enable (ON) or disable (OFF) a subdomain associated to a given adapter in memory. The location code is selected from a list displayed as follows:

```
#ADAPTER CODE       TYPE              SUBDOMAIN   STATE
<adapter code>      <adapter type>    <subdomain> <state>
```

The subdomain associated to the selected adapter is enabled or disabled in memory.

Note: The data base object is not modified.

Dialog Fields

Adapter code Location code of the communications adapter on which the state of the subdomain has to be changed.

***State** Initial state of the subnet entry.
The state is used by the ConnectionLess Network Protocol (CLNP).
If the state is set to 'OFF', the CLNP does not allow to send or receive information using this entry.
The state is not meaningful if the CLNP is not used.
A ring gives the choice between the different values: ON,OFF.

Successful Result

The sudomain state is modified if the user changes the value of a field and a message of confirmation is displayed.

WARNINGS: None.

Error messages: see Error Messages – OSI Stack Configurator on page C-2.

EXAMPLE:

```
Before command completion, additional instructions may appear
below.

Configuration name: OSIdefault

The Subnet 10 state has been modified: old ON, new OFF.
```

Chapter 10. Configuring OSI Stack Using Commands

This chapter describes how to configure the OSI Stack using commands.

Configuring OSI Stack Using Commands: Overview

This section provides information on how to configure the OSI Stack using commands entered at the keyboard. You can find more information in:

- **osiadapt** : Manages the communications adapters and the subnets used by the OSI Stack, see page 10-2.
- **osiconf** : Allows the creation and management of a configuration, see page 10-4.
- **OSIdefault** : Creates the minimum configuration, see page 10-11.
- **osifilter :** Allows the management of security controls, see page 10-12.
- **osiload** : Loads the previously loaded configuration, see page 10-15.
- **osinetbios** : Allows the definition of the NSAP, LSAP and Subdomain Netbios parameters for the current selected configuration, in order to provide the specific Netbios multicast environment. See page 10-16.
- **osinsap** : Allows management of NSAPs in the currently selected configuration, see page 10-18.
- **osirelay** : Allows the addition, removal, or listing of Relays routes for Transport Relay for the current selected configuration. See page 10-20.
- **osirib** : Allows management of the RIB of the currently selected configuration, see page 10-26.
- **osiroute** : Allows the addition, removal or listing of Profiles and Routes for the current selected configuration, see page 10-28.
- **osisnapshot** : Allows display of Dynamic RIB, ES–IS protocol and Memory buffer data concerning the loaded configuration. Also permits resetting of the Static or Dynamic RIB and changing of the subnet status, see page 10-33.
- **osiunload** : De–activates a previously activated configuration. See page 10-35.
- **osix25** : Used to define PVCs in an X.25 network and customize facilities management, see page 10-36.

Note: Command options which are 'static' are indicated **(static only)**. Default is dynamic.

osiadapt Command

Purpose

Manages the OSI configuration data base. It defines the communications adapters (with several lines) used by the OSI Stack.

Syntax

osiadapt [**–v**] **–a** *Adaptercode* ***–s*** *Subdomain* **–t** *State* [**–x** *MaxPDU*]

osiadapt [**–v**] **–c** *Adaptercode* [**–n** *NewAdaptercode*] [**–s** *Subdomain*] [**–t** *State*] [**–x** *MaxPDU*]

osiadapt [**–v**] **–l** [*Adaptertype*]

osiadapt [**–v**] **–r** *Adaptercode* [*Adaptercode...*]

osiadapt [**–v**] **–R** *Adaptercode* [*Adaptercode...*]

Description

The osiadapt command changes the communications adapters or network attachment defined in the current selected configuration.

Four actions are possible: adding adapter (–a), removing adapters (–r or –R), changing adapters (–c), listing configured adapters (–l).

The adapters are uniquely identified by a location code (Adaptercode parameter) which is a string with the format AA–BB or AA–BB–CC or AA–BB–CC–DD.

The different values accepted for the <Adaptercode> parameter are:

`AA-BB :`	`00-00`	`to`	`00-1F`
`AA-BB-CC :`	`00-00-00`	`to`	`00-1F-FF`
`AA-BB-CC-DD :`	`00-00-01-00`	`to`	`00-1F-01-03`
`AA-BB-CC-DD-EE :`	`00-00-01-00-00`	`to`	`00-1F-01-03-00`

Adapters cannot be defined if they are not installed on the machine or if the state of the corresponding device is not Available.

Parameters

Adaptercode Location code of a communications adapter (syntax: AA–BB or AA–BB–CC or AA–BB–CC–DD).

Adaptertype One of the following character strings: ETH, OBIPETH, TKRG, FDDI, X25HIS.

ETH = Ethernet High–performance LAN Adapter

OBIPETH = Standard Ethernet Adapter

TKRG = Token Ring High–performance Network Adapter

FDDI = Network Adapter

X25HIS = X25 HiSpeed WAN Comm. Adapter.

State State of the adapter. Can be up or detach.

MAXPDU Maximum NPDU used (on LAN). The defaults values are the maximum values alllowed, i.e.:
1500 for ETH, OBIPETH,
4000 for FDDI, TKR.

Subdomain Subnet number used by entries in the RIB definition.

Note: The X25HIS (4–port) takes up to 4 lines, each requiring an Adapter configuration.

Flags

–c Used to change a communications adapter in the selected configuration. This option is reserved for a super user.

–a Adds a communications adapter or network attachments in the selected configuration. This option is reserved for super user.

–l Lists the communications adapters defined in the selected configuration. When an Adaptertype is specified, the command displays only the communications adapters of this type.

–r Removes communications adapters. This option is reserved for a super user.

–R Removes communications adapters, and all dependencies. This option is reserved for a super user.

–v Verbose mode.

–x Specifies the maximum NPDU used (on LAN).

–t Specifies the state of the adapter (up/detach).

–s Specifies the subnet number used by entries in the RIB definition.

–n Specifies the new adapter location code.

Examples

1. Add one Ethernet High performance LAN adapter (00–01) .

   ```
   osiadapt -a 00-01 -s 10 -t up -x 1500
   ```

2. Remove the two Ethernet High performance LAN adapters (00–01 and 00–03).

   ```
   osiadapt -r 00-01 00-03
   ```

3. List the TKRG communications adapters.

   ```
   osiadapt -v -l TKRG
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osiadapt	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration). Adapter location code.

Related Information

The **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osirib** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osiconf Command

Purpose

Manages the OSI configuration data base and some parameters in memory if the configuration is the loaded one. It performs two types of action:

- global actions on the configurations (creation, deletion, selection, list, display of configuration parameters),
- management of all the parameters which have a single occurrence within a configuration.

Syntax

Configuration objects management

osiconf –a {**r**|**s**} **–n Name** [**–m Comment**]

osiconf –c Name [**–n Newname**] [**–m Comment**]

osiconf –d Name

osiconf –l [**–v**]

osiconf –r Name

osiconf –s Name

Configurable parameter management

osiconf –i [**–p ConCOPP**] [**–s ConCOSP**] [**–t ConCOTP**] [**–x ConVC**] [**–u usESIS**] [**–e tyESIS**]

(–x ConVC = static only)

osiconf –st SessTimCon SessTimDec [**IndicConTim**]

(IndicConTim = static only)

osiconf –t TransporttimerLAN [**–tw TransporttimerWAN**] **NbretransLAN NbretransWAN [–ta Accept]**

osiconf –e Estimer Istimer

(–e Estimer Istimer = static only)

osiconf –mz MemSize {<**automatic**>|**Size**}

osiconf –x Nodataowner UnknownSPIowner ValueSPI1 OwnerSPI1 ... ValueSPI16

(OwnerSPI16 = static only)

osiconf –lsap LSAP1 [**LSAP2**] ... [**LSAP16**]

Note: Reserved for super–user, except for –d and –l.

Description

Updates the OSI configuration data base.

It is used to perform configuration creation, deletion, selection, list and display.

The creation of a configuration is a mandatory step to be able to configure the OSI Stack. It is possible to create several configurations simultaneously. The configurations may be initialized with default values or by copying an existing configuration.

The list and delete options are available to manage many configurations. But to be able to change the parameters of a specific configuration, it is necessary to select this configuration (using the select option). A configuration remains selected, until being removed or until another one is selected.

The second aim of the osiconf command is the definition of all the parameters of a configuration which have a single instance. For example, the ”maximum number of COTP connections”, or the ”transport timer” have a single value within a configuration.

On the other hand, there may be more than one communications adapter used, or more than one local NSAP. These parameters are not managed by the osiconf command.

Configuration objects management

List all configurations:
Displays for all the defined configurations, the names and the state.

The state is defined for each configuration: it is a combination of idle (not displayed because this state is implicit), selected, loaded. There is no more than one configuration with the state selected or loaded. When the –v option is used, dates and hours of creation or last modification and the comments are displayed too.

The list is in an internal order.

A specific message is displayed when the list is empty and the verbose mode is active.

– Creation of a configuration:
The new configuration may be initialized with default values (using the option ”r”), or by copying all the parameter defined in the current selected configuration (using the option ”s”). The new configuration is automatically selected.

The configurations are identified by their name which must be unique.

– Deletion of a configuration:
Deletes a configuration identified by its name.

Note: A loaded configuration cannot be removed.

– Selection of a configuration:
The selection of a configuration is mandatory in order to change some of the configuration parameters or to create a copy of this configuration.

No more than one configuration can be selected simultaneously.

– Display a configuration:
All the parameters of the given configuration are displayed, even if they are not managed by the osiconf command. It is a verbose description of the configuration.

– Change the name or the comment of a configuration:
This option allow to change the name or the comment of the selected configuration. The new name must remain unique.

Note: The name of a loaded configuration cannot be changed.

Configurable parameters management

All the changes are performed on the selected configuration in the data base (defined configuration), and in memory (loaded configuration) for some parameters if the configuration is the loaded one.

– Definition of common OSI Stack layers parameters:
It is possible to define the maximum number of simultaneous COPP (ConCOPP), COSP (ConCOSP) and COTP (ConCOTP) connections, the maximum number of simultaneous COTP connections on an X25 VC or multiplexing rate (ConVC), and the use (activation) of the ES–IS protocol (usESIS).

Note: The ConCOPP, ConCOSP, ConCOTP and usESIS parameters are updated in memory if the configuration is the loaded one. The ConVC parameter cannot be updated in memory; if the configuration is the loaded one, the command is rejected.

It is possible to define the type of function assumed in the configuration of the ES–IS protocol (tyESIS), for example, End System, Intermediate System, or both simultaneously.

Note: This parameters cannot be updated in memory; if the configuration is the loaded one, the command is rejected.

If options are not set, the current value of the parameter is not changed.

– Definition of the COSP connection timer (SessTimCon):

This option is used to change the COSP connection timer that measures the time interval to establish a connection, in seconds. This option is reserved for advanced configuration and usually there is no reason to change the default value (160 s).

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of the COSP disconnection timer (SessTimDec):

This option is used to change the COSP disconnection timer that measures the time interval to close a connection, in seconds. This option is reserved for advanced configuration and usually there is no reason to change the default value (10 s).

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of the COSP indication connection lifetime (IndicConTim):

This option is used to change the COSP indication connection lifetime that measures the time interval during an indication of connection is maintained (if the receiver is not ready), in seconds. This option is reserved for advanced configuration and usually there is no reason to change the default value (5 s).

Note: This parameter cannot be updated in memory; if the configuration is the loaded one, the command is rejected.

– Definition of transport timer LAN (TransporttimerLAN):

This option is used to change the COTP retransmission timer that measures the time interval (in milli–seconds) before repeating PDUs on LAN networks. This option is reserved for advanced configuration and usually there is no reason to change the default value (10 000 ms).

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of transport timer WAN (TransportimerWAN):

This option is used to change the COTP retransmission timer that measures the time interval (in milli–seconds) before repeating PDUs on WAN networks. This option is reserved for advanced configuration and usually there is no reason to change the default value (10 000 ms).

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of number of retransmissions (NbretransLAN):

This option is used to change the COTP number of retransmissions on LAN networks. This option is reserved for advanced configuration and usually there is no reason to change the default value (10).

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of number of retransmissions (NbretransWAN):

This option is used to change the COTP number of retransmissions on WAN networks. This option is reserved for advanced configuration and usually there is no reason to change the default value (10).

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of Accept Different Responding Address (Accept):

This option is used to define whether an incoming response to a connect request is accepted or not on a different address than the connect address used. Default is Yes.

Note: This parameter is updated in memory if the configuration is the loaded one.

– Definition of ES–IS timer (Estimer and Istimer):

The ES timer (time interval between each ESH PDU) and IS timer (time interval between each ISH PDU), in seconds, may be configured. This option is reserved for advanced tuning (the default value is 180 s).

Note: These parameters cannot be updated in memory; if the configuration is the loaded one, the command is rejected.

– Definition of buffer memory size (Size):

Size – The size of the buffer reserved to process incoming and outgoing frames. When some maintenance tools detect that this value is not the most appropriate one, it is possible to change it. **This option must be used very carefully, because an underestimated value may lead to OSI in framework crash (”Telecom server out of order”)**.

”Automatic” – The size will be calculated automatically at loading time. This is the default value.

Note: This parameter is updated in memory if the configuration is the loaded one.

– Modification of the SPI table:

The Subsequent Protocol Identifier table is used to route incoming calls received on X25 SVCs (Switched Virtual Circuits) to other OSI Stack modules. The default routing is defined by ISO standards, but it is possible to change this definition: the value of the SPI may be changed as well as the destination module (called ”SPI owner”).

Notes:

1. It may occur that the default owner is not available in the OSI framework. The corresponding calls are discarded.
2. These parameters cannot be updated in memory, if the configuration is the loaded one the command is rejected.

Parameters

Comment Character string up to 40 characters including blanks.

Name Character string including ’_’ ’.’ ’/’ and ’\’ (up to 10 characters). Case sensitive.

Newname Character string including ’_’ ’.’ ’/’ and ’\’ (up to 10 characters). Case sensitive.

Nodataowner 4 hexadecimal digits identifying the destination module when there is no user data in the call packet. Default is 0410 (COTP).

ValueSPI<1..16>
Hexadecimal character string (from 1 to 4 digits).

Owner<1..16>
4 hexadecimal digits. Owner<i> identifies the destination module when ValueSPI<i> is detected in the user data of the X25 call packet. Default values are:

no. SPI	ValueSPI	Owner
1:	00	0410 (COTP)
2:	01	3610 (X25.3 access method)
3:	02	3610 (X25.3 access method)
4:	03	0410 (COTP)
5:	81	0F10 (CLNP)
6 to 16:	01	3610 (X25.3 access method)

UnknownSPIowner
4 hexadecimal digits identifying the destination module when the SPI found in the user data of the call packet is not found in the SPI table. Default is 3610 (X25.3 access method)

Note: When defining owner values, it is recommended to use one of the following values: 3610 (X25 access method), 0410 (transport layer), 0f10 (CLNP), 0000 (unknown module).

ConCOPP Integer value. Default is 64. Range is 1 to 1024. This value cannot be greater than the ConCOSP value.

ConCOSP Integer value. Default is 64. Range is 1 to 1024.

ConCOTP Integer value. Default is 64. Range is 1 to 1024.

ConVC Integer value. Default is 64. Range is 1 to 1024. This value can't be greater than the ConCOTP value.

usESIS 0 or 1. Default is 1. Defines if the ES–IS protocol is used.

tyESIS One of the string ES, IS, ES+IS. Default is ES.

SessTimCon Integer. Default is 160. Range is 160 to 20 000. The unit is the second.

SessTimDec Integer. Default is 10. Range is 10 to 20 000. The unit is the second.

IndicConTim Integer. Default is 5. Range is 1 to 20 000. The unit is the second.

TransporttimerLAN
Integer. Default is 3 000. Range is 500 to 30 000. The unit is the millisecond.

TransporttimerWAN
Integer. Default is 15 000. Range is 500 to 60 000. The unit is the millisecond.

NbRetransLAN
Integer. Default is 10. Range is 1 to 15.

NbRetransWAN
Integer. Default is 10. Range is 1 to 15.

Accept Boolean. Default is yes. Value is Yes or No.

Estimer Integer value. Default is 180. range is 1 to 20 000. The unit is the second.

Istimer Integer. Default is 180. Range is 1 to 20 000. The unit is the second.

MemSize Integer. Range for integer is 1000 to 1 000 000. The unit is the Kbyte. Default keyword is ”automatic”.

LSAP1 ... LSAP16
Hexadecimal value in 2 digits. Default is 20. Pre–defined LSAPs ”00”, ”fe” and ”ff” and the LSAP defined with the **osinetbios command** are not allowed.

ribsize Integer value. Default is 256. Range is 256 to 32737.

fixsize Integer value. Default is 200. Range is 1 to ribsize –1.

Flags

Note: There are two levels of flags. The first level is used to specify an action and this flag determines the next sub flags allowed.

–a Creates a new configuration.

–m Attaches a comment to the configuration.

–n Specifies the name of the new configuration.

–c Changes the name and/or the comment of the configuration named Name.

–m Specifies a new comment for the configuration.

–n Specifies the new name of the configuration.

–d Displays all parameters of a configuration identified by its Name.

–l Lists the defined configurations.

– v verbose option (to display the date, time and comment).

–r Removes a configuration identified by its Name.

–s Defines the new selected configuration.

–**i** defines the OSI Stack layer parameters.

–p Defines the ”maximum number of COPP connections”.

–s Defines the ”maximum number of COSP connections”.

–t Defines the ”maximum number of COTP connections”.

–x Defines the ”maximum number of transport connections multiplexed in a VC”. The final multiplexing is the minimum of Nvc and Nbconn.

–u Defines if the ES–IS protocol is used.

–e Defines the ”type of function assumed on the configuration”.

–st Defines the values of the COSP connection timer, disconnection timer and indication connection lifetime.

–t Defines the value of the COTP retransmission timer LAN.

–tw Defines the value of the COTP retransmission timer WAN.

–tr Defines the max number of COTP retransmissions.

–ta Defines acceptance or not of incoming response on another address.

–e Defines the values of the ES and IS timers.

–mz Defines the size of the memory buffer to be allocated.

–x Defines the SPI table of the configuration. The table contains 18 entries (no SPI, unknown SPI, SPI1 to SPI 16). A couple of parameters is provided for each SPI: the value of the SPI and the corresponding destination module.

–lsap Defines up to 16 LSAPs.

Examples

1. Create a configuration named Standard, initialized with default values.

   ```
   osiconf –a r –n Standard
   ```

2. Display the configuration Standard.

   ```
   osiconf –d Standard
   ```

3. Set the maximum number of COTP connection for the selected configuration to 200.

   ```
   osiconf –i –t 200
   ```

4. Activation of the ES–IS protocol:

   ```
   osiconf –i –u 1
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osiconf	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiload** command, **osiunload** command, **osinsap** command, **osirib** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

OSIdefault Command

Purpose

Used to create and define a minimum configuration.

Syntax

OSIdefault

Description

The OSIdefault command allows the system administrator to create and define a minimum configuration which can then be redefined and managed like all others.

Example

See Accessing Minimum Configuration menu, on page 9-36.

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/OSIdefault	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osirib** command, **osisnapshot** command, **osix25** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osifilter Command

Purpose

The OSI Stack provides a mechanism of security access by filters on incoming PDU by several layers. This command allows security on layers COTP, CLTP and CLNP to be enabled or disabled. It allows also to update a table of four fields composed one describing the address type, one the calling address, one the called Tsel and one the called address. These fields are used by layers to filter entry. In case an incoming PDU which does not match a filter: for the layer COTP the CR PDU is not accepted and a DR is responded, in case of Layer CLTP the TSDU is discarded, in case of CLNP the PDU is not relayed.

This command allows the addition, removal, or listing of filters for the current selected configuration. It allows filtering on layers to be enabled or disabled:

COTP filter ”OFF”: No filter are actives on COTP incoming connection establishment

COTP filter ”ON”: Filters are actives on COTP incoming connection establishment

CLTP filter ”OFF”: No filters are actives on CLTP incoming TSDU

CLTP filter ”ON”: Filters are actives on CLTP incoming TSDU

CLNP filter ”OFF”: No filters are actives on CLNP routing PDU

CLNP filter ”ON”: Filters are actives on CLNP routing PDU

Wildcard facilities are allowed for the fields CallingAddr, CalledTsel and CalledAddr.

The following table gives the allowed values for a Filter fields:

Field	**Value**	**Comment**
AddrType	NSAP, LAN, SVC, PVC	
CalledAddr	wildcard	All called Addresses will match this field.
	no wildcard	Only called addresses with same value as the field will match this field. if AddrType is LAN address is a MAC Address if AddrType is NSAP address is a NSAP Address if AddrType is SVC address is a SVC Address if AddrType is PVC address is a PVC Name
CalledTsel	wildcard	All called TSELs will match this field.
	no wildcard	Only called TSELs with same value as the field will match this field.
CallingAddr	wildcard	All calling Addresses will match this field
	no wildcard	Only calling addresses with same value as the field will match this field. if AddrType is LAN address is a MAC Address if AddrType is NSAP address is a NSAP Address if AddrType is SVC address is a SVC Address

The NULL TSEL value is allowed for the TSEL field of one Filter.

Syntax

osifilter –a FilterName <CalledTsel> <AddrType> <CalledAddr> <CallingAddr>

osifilter –l

osifilter [–R] [–T <cotp>] [–L <cltp>] [–N <clnp>]

osifilter –r FilterName [FilterName ...]

Description

Note: This command applies to the selected configuration.

When a Filter entry is added (–a) or removed (–r) the data base is updated, then if the selected configuration is the loaded one, the table in memory is updated from the data base.

At loading time the Filter table is loaded in memory from the data base.

Note: The time to add or remove a Filter entry depends on the number of entries in the data base, so it is recommended to add or remove Filters entries with the configuration unloaded.

The Filter entries defined in a configuration must have a unique occurrence of a key composed with the **AddrType**, **CallingAddr, CalledTsel** and **CalledAddr** parameters.

Parameters

cotp, **cltp**, **clnp** value 0 or 1

FilterName alphanumeric character string including '_' '.' '/' and '\' (up to 10 characters) The Filters defined in a configuration must have a unique FilterName and a unique occurrence of a key composed with the **AddrType**, **CallingAddr, CalledTsel** and **CalledAddr** parameters.

CalledTsel hexadecimal string up to 64 digits, even number, or "0" or "–". (default "–").

AddrType string "NSAP" or "LAN" or "SVC" or "PVC" (lowercase allowed).

CalledAddr if AddrType is "NSAP", hexadecimal string up to 40 digits, even number.

if AddrType is "LAN", hexadecimal string of 12 digits.

if AddrType is "SVC", decimal string up to 15 digits.

if AddrType is "PVC", a name in a string of 1 to 8 alphanumeric character string including '_' '.' '/' and '\'.

CallingAddr if AddrType is "NSAP", hexadecimal string up to 40 digits, even number.

if AddrType is "LAN", hexadecimal string of 12 digits.

if AddrType is "SVC", decimal string up to 15 digits.

if AddrType is "PVC", this field is irrelevant.

Note: The NULL TSEL is defined with the string "0" (zero), and all wildcard values with the string "–".

Flags

–R	Reads layer status filtering
–T	Change COTP layer status filtering. Reserved for super–user.
–L	Change CLTP layer status filtering. Reserved for super–user.
–N	Change CLNP layer status filtering. Reserved for super–user.
–a	Adds a single Filter entry. Reserved for super–user.
–l	Lists the defined Filters entries.
–r	Removes a list of Filters entries. Reserved for super–user.

Return Values

Upon successful completion, a value of 0 (zero) is returned.

If an error occurs, the data base is not modified, a message is displayed on stdout and –1 value is returned. A distinguished *errno* value is also set.

Examples

1. List the defined Filters entries

```
osifilter -l
```

2. Enable filtering on COTP

```
osifilter -T 1
```

3. Add a Filter entries:

   Accept only Incoming PDU with called TSAP 0x5467, calling NSAP 490502 and Called NSAP 0x490101:

```
osifilter -a Fil100 5467 nsap 490501 490102
```

   Accept only Incoming connection with any called TSEL and Any Lan Addresses from 0x010203040506

```
osifilter -a Fil101 "-" lan "-" 010203040506
```

   Accept all incoming connection from X25.

```
osifilter -a Fil102 "-" svc "-" "-"
```

   Accept all incoming connection on PVC name "bibi"

```
osifilter -a Fil103 "-" pvc bibi
```

4. Remove a defined Filter entry

```
osifilter -r Fil103
```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osifilter	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osirib** command, **osinetbios** command, **osiroute** command and **osirelay** command.

osiload Command

Purpose

Activates a previously generated configuration.

Syntax

osiload

Description

This command load in memory (activate) a configuration previously selected. If another configuration is already loaded it must be unloaded previously. When there is no other configuration loaded, the OSI driver, the static RIB and the initialization shell are activated by the command. The daemon **osilinkd** is started to complete the loading.

Note: If the machine is rebooted, the currently loaded configuration is automatically activated during the boot phase.

Examples

1. Load a selected configuration.

   ```
   osiload
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

OSICONF.cat	Contains the message catalog file.
/usr/bin/osiload	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiconf** command, **osiunload** command, **osinsap** command, **osirib** command, **osisnapshot** command, **osix25** command **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osinetbios Command

Purpose

Allows the definition of the NSAP, LSAP and Subdomain Netbios parameters for the current selected configuration, in order to provide the specific Netbios multicast environment.

Reserved for a super–user.

Syntax

osinetbios [–nn NSAP] [–nl LSAP] [–ns Subdomain]

osinetbios –s

Description

Note: This command is only available on a static (not loaded) selected configuration.

Used to define the NSAP, LSAP and Subdomain Netbios parameters, or to show the previously defined or default ones (–s).

The NSAP syntax (AFI, IDI and DSP fields) is not verified.

Only one set of NSAP, LSAP and Subdomain Netbios values are available in a configuration; each time the command is executed, the new values replace the precedent ones.

The Netbios NSAP and LSAP must be unique in the configuration.

The Netbios NSAP has to be different of all Local NSAPs defined (see the **osinsap** command, on page 10-18).

The Netbios LSAP has to be different from all defined LSAPs for the NULL CLNP profile (see the **osiconf** command, on page 10-4), and different from the predefined LSAPs: ”00”, ”fe” and ”ff”.

The Subdomain needs to be previously defined (see the **osiadapt** command, on page 10-2), on a non X.25 communications adapter, in the configuration. The default is the lowest number between the LAN subdomain defined and is initialized at loading time.

Parameters

NSAP	hexadecimal string: the number of digits is even, between 2 and 40.
	Default value: ”49000009006a00010001”.
LSAP	hexadecimal string of 2 digits. Default value is: ”ec”.
Subdomain	hexadecimal value (not null) up to 8 digits.
	Default value: lowest number between the LAN subdomains defined.

Flags

–nn	Defines the Netbios NSAP.
–nl	Defines the Netbios LSAP.
–ns	Defines the Netbios Subdomain.
–s	Show the defined Netbios parameters (NSAP, LSAP and Subdomain).

Return Values

Upon successful completion, a value of 0 (zero) is returned.

If an error occurs, the data base is not modified, a message is displayed on stdout and –1 value is returned.

Examples

1. Definition of the Netbios NSAP

   ```
   osinetbios -nn 49010203
   ```

2. Define all Netbios parameters

   ```
   osinetbios -ns 200 -nl 2f -nn 49010203
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osinetbios	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osirib** command, **osiroute** command, **osirelay** command and **osifilter** command.

osinsap Command

Purpose

Allows the addition, removal, or listing of NSAPs for the current selected configuration.

Syntax

osinsap [–v] –a NSAP [local|group]

osinsap [–v] –l

osinsap [–v] –L

osinsap [–v] –r NSAP ...

Description

Used to define the Local NSAPs. The actions (adding an NSAP, listing defined NSAPs, with or without their types, or removing NSAPs) apply to the selected configuration.

It is possible to define up to 32 NSAPs in each configuration. The NSAPs defined in a configuration must be all different. The syntax is not verified.

Note:

1. For the option ”add an NSAP” if the selected configuration is the loaded one, the NSAP is also added in memory.
2. For the option ”add an NSAP”, if neither ’local’ or ’group’ is provided, ’local’ is assumed,
3. For the option ”remove an NSAP” if the selected configuration is the loaded one, the command is applied to the memory.

Parameters

NSAP	hexadecimal string: the number of digits is even, between 2 and 40.

Flags

–v	Verbose mode.
–a	Adds a single NSAP. Reserved for super–user. ’local’ is an NSAP used by ES–IS, while ’group’ is not.
–l	Lists the defined NSAPs (without printing NSAP types).
–L	Lists the defined NSAPs and their types.
–r	Removes a list of NSAPs. Reserved for super–user.

Examples

1. List the defined NSAPs

```
osinsap –v –l
Configuration name: OSIdefault
  NSAP NAME
– – – – – – – – – – – – – – – – – – – – –
49000009006a00010001
30303030
```

2. Add the NSAP DEAD (it is not an ISO NSAP)

```
osinsap –a DEAD local
```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osinsap	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osirib** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osirelay Command

Purpose

This command allows the addition, removal, or listing of Relays routes for this TSEL Transport Relay for the current selected configuration.

The OSI Stack provides a LAN–WAN Transport relay operating over NULL CLNP profile, i.e. a relay based on TSELs. It allows to relay 2 Transport connections, an incoming one (indicated by COTP) and an outgoing one (requested by the relay to COTP). This assures 2 End Systems which resides on separated networks to communicate through the relay.

A Relay route is defined by the following relation:

Incoming address	Outgoing address	Additional info
InTselEd InTselIng InAddrType InAddrIng	OutTselEd OutTselIng OutAddrType OutAddrEd OutAddrIng OutLsapEd	InNegoClass

[**InTselEd**, **InTselIng**, **InAddrType** and **InAddrIng**] is the key to access the relay data base to find a matching route, which gives the outgoing address [OutTselEd, OutTselIng, OutAddrEd,OutAddrIng], and the Transport class [InNegoClass] when possible in the incoming X25 connections, to perform the relaying function.

Wildcard facilities are allowed for all the key fields. One field to wildcard value means that the route is defined for all the possible values of the field.

Conflicts between several route definitions are solved by the following precedence rule, from higher to lower priority:

InTselEd	InTselIng	InAddrIng
No Wildcard	No Wildcard	No Wildcard
No Wildcard	No Wildcard	Wildcard
No Wildcard	Wildcard	No Wildcard
No Wildcard	Wildcard	Wildcard
Wildcard	No Wildcard	No Wildcard
Wildcard	No Wildcard	Wildcard
Wildcard	Wildcard	No Wildcard
Wildcard	Wildcard	Wildcard

Wildcard facilities are allowed for the fields OutTselEd, OutTselIng and for the field OutAddrIng only if it is a NSAP.

If the field OutTselEd (resp. OutTselIng) is set to wildcard value, the called (resp. calling) TSEL of the outgoing connection is set to the value of the called (resp. calling) TSEL present into the incoming connection.

If the field OutLsapEd is set to the wildcard value, the called LSAP of the outgoing connection is set to the value of the first LSAP in the list defined for NULL CLNP profiles.

The field **OutAddrEd** is the key to access the Profile data base which provides the Transport parameters associated to the outgoing route.

If the NSAP provided into OutAddrIng field is set to wildcard value, the value of the outgoing calling NSAP is set to the value of one of the local NSAPs.

The OutAddrIng field is not significant when OutAddrEd is an X25 PVC name.

The InNegoClass is significant only for incoming X25 connections. The relay tries to negotiate the incoming connection class to the InNegoClass proposed value, if the preferred and alternate incoming connection class allows it. When the negotiation is not allowed, the preferred class is accepted. Other Transport parameters may be negotiated to produce a valid set of parameters.

The following table gives the allowed values for a Relay route fields:

Field	Value	Comment
InTselEd	wildcard	All incoming called TSELs will match this field.
	no wildcard	Only incoming called TSELs with same value as the field will match this field.
InTselIng	wildcard	All incoming calling TSELs will match this field.
	no wildcard	Only incoming calling TSELs with same value as the field will match this field.
InAddrIng	wildcard	The type of the address is not relevant for the route; all incoming addresses will match this field, whatever their type.
	no wildcard	Only incoming calling addresses with same value as the field will match this field.
OutTselEd	wildcard	The outgoing called TSEL will be the incoming one.
	no wildcard	The outgoing called TSEL will be the field value.
OutTselIng	wildcard	The outgoing calling TSEL will be the incoming one.
	no wildcard	The outgoing calling TSEL will be the field value.
OutAddrEd	no wildcard	The outgoing called address type and value will be those of the field.
OutAddrIng or OutSubdIng	wildcard if Out-AddrType is CL or CLCO	The outgoing calling address will be replaced by the first local NSAP
	otherwise no wildcard	The outgoing calling address type and value will be the field ones. This field is not significant if Out-AddrEd is an X25 PVC name.
InNegoClass	0,2,3 or 4	Significant only for incoming X25 connections. The relay reduces the incoming connection class to the field value if the incoming preferred an alternate classes allows this, else the relay accept the preferred class.
OutLsapEd	wildcard	The outgoing called LSAP will be the first LSAP of the NULL CLNP LSAP list.
	no wildcard	The outgoing called LSAP will be the field value.

The NULL TSEL value is allowed for all TSEL fields of one Relay route.

Syntax

osirelay **–a RelayName** [**–ite InTselEd**] [**–iti InTselIng**]
[**–iat InAddrType**] [**–iai InAddrIng**] [**–ic InNegoClass**]
[**–ote OutTselEd**] [**–oti OutTselIng]**
–oad OutAddrType OutAddrEd {**OutAddrIng** | **OutSubdIng**}
[**–ole OutLsapEd**]

osirelay **–l**

osirelay **–r RelayName** [**RelayName** ...]

Description

Note: This command applies to the selected configuration.

When a Relay route entry is added (–a) or removed (–r) the data base is updated, then if the selected configuration is the loaded one, the table in memory is updated from the data base.

At loading time the Relay route table is loaded in memory from the data base.

Note: The time to add or remove a Relay route entry depends on the number of entries in the data base, so it is recommended to add or remove Relays routes entries with the configuration unloaded.

The Relay routes defined in a configuration must have a unique occurrence of a key composed with the **InTselEd**, **InTselIng**, **InAddrType** and **InAddrIng** parameters.

Parameters

RelayName alphanumeric character string including '_' '.' '/' and '\' (up to 10 characters) The Relay routes defined in a configuration must have a unique RelayName and a unique occurrence of a key composed with the **InTselEd**, **InTselIng**, **InAddrType** and **InAddrIng** parameters.

InTselEd hexadecimal string up to 64 digits, even number, or ”0” or ”–”. (default ”–”).

InTselIng hexadecimal string up to 64 digits, even number, or ”0” or ”–”. (default ”–”).

InAddrType string ”NSAP” or ”LAN” or ”SVC” or ”PVC” (lowercase allowed).

InAddrIng if InAddrType is ”NSAP”, hexadecimal string up to 40 digits, even number.

if InAddrType is ”LAN”, hexadecimal string of 12 digits.

if InAddrType is ”SVC”, decimal string up to 15 digits.

if InAddrType is ”PVC”, a name in a string of 1 to 8 alphanumeric character string including '_' '.' '/' and '\'.

If InAddrIng is set to the wildcard value (”–”), then InAddrType is not significant.

InNegoClass character ”0”, ”2”, ”3” or ”4”. (default ”4”). InNegoClass is significant only when InAddrType is ”SVC” or ”PVC”, and when InAddrType is ”NSAP” and the connection is received on the profile: COTP over CLNP on LAN and COTP overs CONS on WAN.

OutTselEd hexadecimal string up to 64 digits, even number, or ”0” or ”–”. (default ”–”).

OutTselIng hexadecimal string up to 64 digits, even number, or ”0” or ”–”. (default ”–”).

OutAddrType string ”SVC”, ”LAN”, ”CL”, ”LANC”, ”CLCO” or ”PVC” (lowercase allowed). This parameter is mandatory.

”SVC”: for a profile X25 SVC

”LANC”: for a profile LAN Configurable

”CL”: for a profile NSAP/CLNP

”LAN”: for a profile SNPA/FE

”CLCO”: for a profile ConnectionLess/Connection Oriented

”PVC”: for a profile X25 PVC

If OutAddrType is ”PVC”, OutAddrIng or OutSubdIng are not significant; The PVC name will be provided in the OutAddrEd parameter.

OutAddrEd if OutAddrType is ”SVC”, decimal string up to 15 digits.

if OutAddrType is ”LANC” or ”LAN”, hexadecimal string of 12 digits.

if OutAddrType is ”CL” or ”CLCO”, hexadecimal string up to 40 digits, even number.

if OutAddrType is ”PVC”, a name in a string of 1 to 8 alphanumeric characters including ’_’ ’.’ ’/’ and ’\’.

OutAddrIng if OutAddrType is ”CL” or ”CLCO” this parameter is significant instead of OutSubdIng parameter; hexadecimal string up to 40 digits, or ”–”, (default ”–”).

if OutAddrType is ”PVC” this parameter is not significant.

OutSubdIng if OutAddrType is ”SVC”, ”LANC”, or ”LAN”, this parameter is significant instead of OutAddrIng parameter; an hexadecimal value, not null, (up to eight digits) corresponding to a defined Subdomain.

if OutAddrType is ”PVC” this parameter is not significant.

OutLsapEd if OutAddrType is ”LANC”; hexadecimal string of 2 digits or ”–”, (default ”–”), else this parameter is not significant.

Flags

–a Adds a single Relay route entry. Reserved for super–user.

- **–ite** defines the incoming called TSEL
- **–iti** defines the incoming calling TSEL
- **–iat** defines the incoming calling Network Address type
- **–iai** defines the incoming calling Network Address
- **–ic** defines the incoming calling class negotiation on an X25 connection
- **–ote** defines the outgoing called TSEL
- **–oti** defines the outgoing calling TSEL
- **–oad** defines the outgoing Network type, outgoing called Network Address, and outgoing calling Network Address or subdomain.
- **–ole** defines the outgoing called LSAP.

–l Lists the defined Relays routes entries.

–r Removes a list of Relays routes entries. Reserved for super–user.

Return Values

Upon successful completion, a value of 0 (zero) is returned.

If an error occurs, the data base is not modified, a message is displayed on stdout and –1 value is returned.

Examples

1. List the defined Relays routes

   ```
   osirelay -l
   ```

2. Add a Relay route

   ```
   osirelay -a Rel100 -ite 112233 -iti - -iat NSAP -iai 49000099
   -oad lan 010203040506 200
   ```

3. Remove a defined Relay route

   ```
   osirelay -r Rel100
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osirelay	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osirib** command, **osiroute** command and **osifilter** command.

osirib Command

Purpose

Manages the Routing Information Base (RIB) of the current selected configuration.

Syntax

osirib [–v] –a RemoteNSAP RemoteNSAPtype [–m RemoteNSAPmask] Network
[–t Connectiontype] RemoteSNPAtype RemoteSNPA Subdomain Priority RIBpart
State Msglifetime [–x CUG]

osirib [–v] –c Internid RemoteNSAP RemoteNSAPtype [–m RemoteNSAPmask] Network
[–t Connectiontype] Remote SNPAtype RemoteSNPA Subdomain Priority RIBpart
State Msglifetime [–x CUG]

osirib [–v] –l

osirib [–v] –r Internid [Internid] ....

Description

Allows the definition of the static RIB of the current selected configuration.

The RIB stores routing information to be able to establish connections using two lots of addressing information: the local NSAP and the remote NSAP. To establish the connection, the remote NSAPs must be reached.

The static RIB is fully described in Static Routing Information Base, on page 8-16.

Parameters

Connectiontype
PVC or SVC.

CUG Integer value in the range from 0 to 9999. The value 0 means that the facility is not used. Default is 0.

Internid Integer value.

Part STATICF (fixed part), or STATICV (variable part).

Msglifetime Integer value in the range from 1 to 65535.

Network ETH (Ethernet), TKRG (Token Ring), FDDI, X25.

Priority Integer value in the range from 1 to 65535.

RemoteNSAP NSAP (hexadecimal string with an even number of digits between 2 and 40).

RemoteNSAPmask
String with ’0’ and ’f’ or ’F” characters (0*f*)*. The length is either 0 (no mask) or the length of the RemoteNSAP parameter.

RemoteNSAPtype
NSAP or MSDSG (significant for relaying only) use of CLNP relay or MSDSG relay.

RemoteSNPA Depends of the value of the Network, Connectiontype and RemoteSNPAtype parameters. It may be either a LAN address (12 hexadecimal digits), or an X25 address (between 1 and 15 hexadecimal digits), or a PVC name, or an NSAP name (hexadecimal string, even number of digits between 2 and 40), or a LAN mask (with 12 F characters), or a WAN mask (with up to 15 F characters). See previous explanation for the syntax.

RemoteSNPAtype
SNPA, MASK or NSAP

State ENABLE or LOCKED.

Subdomain Hexadecimal value (up to 4 bytes). Not null.

Flags

–a Adds a new RIB entry.

–c Changes the RIB entry identified by the parameter Internid.

–l Lists all defined RIB entries with all the parameters.

–r Removes a list of RIB entries.

–v Verbose mode.

Examples

1. Create a RIB entry to access the remote NSAP 3140. The remote SNPA is 80 (X25 address), the local sub domain is 200. The entry belongs to the fixed part of the RIB. CUG is used.

```
osirib –a 3140 NSAP X25 –t SVC SNPA 80 200 1 STATICF
ENABLE 15 –x 4
```

2. Create a RIB entry to access the remote NSAPs 4250 and 4360. The remote SNPAs are respectively 450 and 460 (X25 address), the local sub domain is 300. The entry belongs to the variable part of the RIB. 10 CUG are allowed.

```
osirib –a 4000 NSAP –m f000 X25 –t SVC MASK f0ff 300 1 STATICV
ENABLE 15
```

3. Create a RIB entry to access the remote NSAP 5000 using the PVC TRAN0101; the local sub domain is 400. The entry belongs to the variable part of the RIB.

```
osirib –a 5000 NSAP  X25 –t PVC SNPA TRAN0101 400 1 STATICV
ENABLE 15
```

4. Create a RIB entry to access the remote NSAP whose syntax is 33<remote SNPA>; the local sub domain is 500 (Ethernet adapter). The entry belongs to the fixed part of the RIB.

```
osirib –a 33000000000000 NSAP –m ff000000000000 ETH MASK
00ffffffffffff 500 1 STATICF ENABLE 15
```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osirib	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration). Adapter location code.

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osiroute Command

Purpose

Allows the addition, removal or listing of Profiles and Routes for the current selected configuration.

A Profiles and Routes data base allows the OSI Stack to retrieve Transport parameters from the called address (NSAP or SNPA), instead of receiving them directly through the stack interfaces, at the connection establishment phase.

The Transport configurable parameters are:

Transport class,

alternate Transport class,

use of expedited data,

use of flow control,

use of extended format,

use of checksum,

use of Alternate Retransmit Algorithm,

priority of the connection,

credit,

TPDU size

and, for X25 type networks,

CUD,

CUG facility.

The Profiles must be defined before their associated Routes.

When a Profile is removed, the associated Routes are also removed.

Syntax

osiroute –lp

osiroute –lr [ProfileName]

osiroute –ap ProfileName –tc Tclass [–ta Taltclass] [–ex UseExpData] [–fl UseFlowCtrl]
[–et UseExtendFmt] [–ck UseChecksum] [–pr Priority] [–cr Credit] [–sz TPDUsize]
[–cd CUD] [–cg CUG] [–ea UseAltRetryAlgo]

osiroute –ar RouteName ProfileName RouteType RouteValue

osiroute –rp ProfileName [ProfileName ...]

osiroute –rr RouteName [RouteName ...]

osiroute –sr RouteName

Description

Note: This command applies to the selected configuration, the data base and the memory if the configuration is loaded.

For the options ”add (–ap, –ar) and remove (–rp, –rr)”, the data base is updated, then if the selected configuration is the loaded one, the Profiles table and the Routes table are updated in memory.

When the stack is loaded, the Profiles table and the Routes table are loaded in memory.

Note: The time to add or remove a Profile or Route entry depends on the number of entries in the data base, so it is recommended to add or remove Profile and Route entries with the configuration unloaded.

The command verifies the coherence between the different values in a Profile or in a Route and its Profile.

The values of the parameters not given in the command are added according to the class.

Tclass		**2**	**3**	**4**
Taltclass	–1	0	0	–1
UseExpData	0	1	1	1
UseFlowCtrl	0	1	1	1
UseExtendFmt	0	0	0	0
UseChecksum	0	0	0	0
UseAltRetryAlgo	not applicable	not applicable	not applicable	0
Priority	DFLT	DFLT	DFLT	DFLT
Credit	not applicable	15	15	15
TPDUsize	2048	8192	8192	8192
CUD	0	0	0	0
CUG	–1	–1	–1	–1

The default value ”–1” means <none>.

The user can define two default Profiles. Their names are reserved and cannot be used for other Profiles: ”**LANprofile**” and ”**WANprofile**”.

The default profiles can be added and removed like all other profiles; these profiles can be added with the command:

```
osiroute –ap LANprofile
osiroute –ap WANprofile
```

if the default parameters here shown are the appropriate for the configuration.

These profiles are used when the route is not found in the data base. The WANprofile is used for routes ”SVC” or ”PVC”; the LANprofile is used for all other routes.

LANprofile default values:

```
Tclass           ”4”
Taltclass        ”–1”
UseExpData       ”1”
UseFlowCtrl      ”1”
```

UseExtendFmt	”0”
UseChecksum	”0”
UseAltRetryAlgo	”0”
Priority	”DFLT”
Credit	15
TPDUsize	8192
UD	0
CUG	–1

WANprofile default values:

Tclass	”2”
Taltclass	”0”
UseExpData	”1”
UseFlowCtrl	”1”
UseExtendFmt	”0”
UseAltRetryAlgo	”0”
UseChecksum	”0”
Priority	”DFLT”
Credit	15
TPDUsize	8192
CUD	0
CUG	–1

Parameters

ProfileName alphanumeric character string including '_' '.' '/' and '\' (up to 10 characters). The ProfileName defined must be unique in the configuration. The names **LANprofile** and **WANprofile** are reserved.

Tclass character string ”0”, ”2”, ”3” or ”4”. This value needs to be ”4” If the ProfileName is **WANprofile**.

Taltclass character string ”0”, ”2”, ”3”, ”4” or ”–1”. This value needs to be lower or equal to the **Tclass** value.

UseExpData ”0” or ”1” character flag. This value needs to be ”0” if the **Tclass** value is ”0”. If this value is ”1” and the value of **Tclass** is ”2” then the UseFlowCtrl needs to be ”1”.

UseFlowCtrl ”0” or ”1” character flag. This value needs to be ”0” if the **Tclass** value is ”0”, ”0” or ”1” if the **Tclass** value is ”2” and ”1” if the **Tclass** value is ”3” or ”4”. This value needs to be ”1” if the value of **Tclass** is ”2” and UseExpData is ”1”.

UseAltRetranAlgo
”0” or ”1” character flag. This value needs to be ”0” if the **Tclass** value is not ”4”. If this value is ”1” and the value of **Tclass** is ”4” then the UseAltRetranAlgo Forced the use of an Alternate Algorithm of COTP Retransmission which retransmit all the PDU not ACK in the window since the first not ACK instead of the first not ACK.

UseExtendFmt ”0” or ”1” character flag. This value needs to be ”0” if the **Tclass** value is ”0”, ”2” or ”3”.

UseChecksum "0" or "1" character flag. This value needs to be "0" if the **Tclass** value is "0", "2" or "3".

Priority character string ["LOW", "MID", "HIGH", "TOP","DFLT"] (lowercase allowed).

Credit decimal value [1..15]. This value is not significant if UseFlowCtrl is "0".

TPDUsize decimal value [128,256,512,1024,2048,4096,8192]. This value needs to be lower or equal to 2048 if the **Tclass** value is "0".

CUD hexadecimal string up to 8 digits. If the number of digits is odd a 0 digit is added at the beginning of the string. (Only for X25 Networks).

CUG decimal value [–1..9999]. [0..99] for the Basic format class A, and [100..9999] for the Extended format class B. The value –1 means no CUG facility. (Only for X25 Networks).

RouteName alphanumeric character string including '_' '.' '/' and '\' (up to 10 characters). The RouteName defined must be unique in the configuration.

RouteType character string ["NSAP", "LAN", "SVC", "PVC"] (lowercase allowed). If this value is "LAN" the associated Profile needs to have a **Tclass** value to "4".

RouteValue if RouteType is "NSAP", hexadecimal string up to 40 digits, even number.

if RouteType is "LAN", hexadecimal string of 12 digits.

if RouteType is "SVC", decimal string up to 15 digits.

if RouteType is "PVC", a name in a string of 1 to 8 alphanumeric character string including '_' '.' '/'.

Flags

–lp Lists the defined Profiles.

–lr Lists the defined Routes or the routes associated to a given Profile.

–ap Addition of a new Profile. Reserved for super–user.

–tc defines the Transport class

–ta defines the alternate Transport class

–ex defines the use of "expedited data"

–fl defines the use of "flow control"

–ea defines the use of "Alternate Retransmission Algorithm"

–et defines the use of "extended control"

–ck defines the use of "checksum"

–pr defines the Priority value

–cr defines the Credit value

–sz defines the TPDU max size

–cd defines the CUD (or SPI) value for X25 Network

–cg defines the CUG facilitie value for X25 Network

–ar Addition of a new Route. Reserved for super–user.

–rp Removes Profiles and associated Routes. Reserved for super–user.

–rr Removes Routes. Reserved for super user.

–sr Shows a Route and its Profile.

Return Values

Upon successful completion, a value of 0 (zero) is returned.

If an error occurs, the data base is not modified, a message is displayed on stdout and –1 value is returned.

Examples

1. List the defined Routes on a given Profile

   ```
   osiroute –lr PROF200
   ```

2. Add a Profile

   ```
   osiroute –ap PROF500 –fl 0 –sz 2048 –tc 2
   ```

3. Add a Route

   ```
   osiroute –ar Rout PROF500 NSAP 4901010199
   ```

4. Add a Route

   ```
   osiroute –ar Rout PROF500 PVC pvc001
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osiroute	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration). Adapter location code.

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osirib** command, **osirelay** command, **osinetbios** command and **osifilter** command.

osisnapshot Command

Purpose

Allows some information to be shown and special actions to be performed on the loaded configuration (i.e. memory)

Syntax

osisnapshot –ln

osisnapshot –lo

osisnapshot –lw

osisnapshot –ld

osisnapshot –ls

osisnapshot –sd Internid

osisnapshot –td

osisnapshot –ts

osisnapshot –de

osisnapshot –dm

osisnapshot –cs Adaptercode Newstate

Description

Shows information in a loaded configuration about the Dynamic RIB, the ES_IS protocol status and the Memory buffer status.

Allow to reset the Static or the Dynamic RIB.

The command will fail if the OSI Stack has not been Loaded.

The reset of the static RIB (–ts option) flushes the static RIB information loaded in memory and reloads the fixed part of the static RIB.

Two list options allow to list the dynamic RIB, and the wait entries (remote NSAPs which are required but not yet resolved by the **osiribd** daemon) see Routing Framework and Routing Information Base, on page 3-10.

On Dynamic RIB only the list, show and reset actions are possible.

The reset of the dynamic RIB (–td option) flushes the information stored in the dynamic RIB.

Parameters

Internid	Integer value. Internid is displayed by the option –ls or –ld.

Flags

–ln	Displays the local NSAPs and their types.
–lo	Displays the Adapters addresse, type, subdomain, location and state.
–lw	Displays the waiting RIB entries.
–ld	Displays a summary of each dynamic RIB entry.
–ls	Displays all static RIB entries.
–sd	Shows an entry in the dynamic RIB.
–td	Resets dynamic RIB. Only for super–user.

–ts Resets static RIB. Only for super–user.

–de Shows the type and the status of the ES–IS protocol.

–dm Shows the Memory buffer status.

–cs Changes the subnet state ('ON' or 'OFF').

Examples

1. List the Dynamic RIB entries:

```
osisnapshot -ld
```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osisnapshot	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).
Adapter location code.

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osirib** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osiunload Command

Purpose

De–activates a previously activated configuration.

Syntax

osiunload

Description

De–activates a configuration which has been previously activated using the command osiload. This configuration is not necessarily the current selected one. The configuration can be de–activated only if the OSI driver is not busy (i.e. is not used by any application). If a configuration is active, a signal is sent to the **osilinkd** daemon which stops the OSI framework, and update the configuration state (set to not loaded).

Examples

1. De–activate the active configuration.

   ```
   osiunload
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

OSICONF.cat	Contains the message catalog file.
/usr/bin/osiunload	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).
Adapter location code.

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osinsap** command, **osirib** command, **osisnapshot** command, **osix25** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

osix25 Command

Purpose

Used to define PVCs and to customize the management of the facilities.

Syntax

osix25 –C Adaptercode PVCname Channel Owner

osix25 –c Adaptercode [–s Site] [–o0 Owner] ... [–o255 Owner]

osix25 –f Adaptercode [–f1 0|1] [–f2 0|1] [–f3 0|1] [–f4 0|1] [–f5 0|1] [–f6 0|1|2]
[–f7 Transitdelay] [–f8 Endtoend] [–g5 EndtoEnd]

osix25 –l Adaptercode

osix25 –r Adaptercode PVCname [PVCname....]

osix25 –s Adaptercode PVCname Channel Owner

Description

The osix25 command has three mains functions:

PVC management,

facilities management,

check of defined information.

The command impacts only the OSI configuration data base of the current selected configuration.

PVC management

For each X25 communications adapter defined in the selected configuration, the maximum number of defined PVCs depends upon the communications adapter used and device driver used.

Each PVC is identified by a Name (character string).

A PVC is defined if it has a not null owner. The owner is the identification of an OSI Stack module that will receive all the incoming data of the PVC. Typical modules are transport and X25 access method.

There are two syntaxes to add or change a PVC:

with the flag **–c**

With this syntax is possible to define up to 256 PVCs. Each PVC name is identified by a character string with the following syntax: <Site>01<PVC number>. The Site is a character string that uniquely identifies an X25 communications adapter (up to 4 characters, case sensitive). The next parameter is allays 01, because there is a single line per adapter. The last parameter <PVC number> is a two digits hexadecimal number between 00 and FF (in the command syntax is 0 to 255); so by this mean only can be defined the first 256 PVC.

with the flag **–C**

With this syntax the limit is defined by the communications adapter used.

Facilities management

The aim is to customize the management of the X25 facilities: the OSI Stack is able to add facilities in the call and call confirm packets (of course if the X25 communications adapter is configured to support these facilities).

Parameters

Adaptercode Communications adapter location code (syntax: AA–BB [–CC [–DD [–EE]]]: identification of the X25 communications adapter on which PVC will be defined.

Transitdelay Integer value in the range from 0 to 65534, specifying a timer in milliseconds. 65335 means facility not forced. Default is 65535.

Endtoend Integer value in the range from 0 to 65534, specifying a timer in milliseconds. 65335 means facility not forced. Default is 65535.

PVCname A string of up to 10 alphanumeric characters (including '_' '.' '/' and '\'.

Channel Is the Logical Channel Number (LCN). Its value is in the range of 0 to 4095.

Owner Is a value of 1 to 4 hexadecimal digits identifying an OSI Stack module. It is recommended to choose one of the following values: 0000 (PVC not defined), 0410 (transport layer), 3610 (X25 access method).

Site Character string (up to 4 characters) used as a prefix for the PVC name. Uniquely identifies an X25 communications adapter. The default is 'SITE'

0..255 Is the Logical Channel Number (LCN). Its value is in the range of 0 to 255.

Flags

–C Defines a PVC. Reserved for super–user.

–c Defines several PVCs with an old syntax command. Only the first 255 pvc can be defined with this syntax. Reserved for super–user.

–s Optional Site name. If not specified, the Site is set to the default value.

–o0..–o255 Defines the owner of the PVC number o<i>.

–f Changes the default management of the facilities. Reserved for super–user. If a sub flag is not specified the corresponding facility is not changed.

–f1 Defines if the facility "negotiation of the packet size" is added in the call packet (0 not added, 1 added). Default is 0.

–f2 Defines if the facility "negotiation of the window size" is added in the call packet (0 not added, 1 added). Default is 0.

–f3 Defines if the facility "negotiation of the throughput class" is added in the call packet (0 not added, 1 added). Default is 0.

–f4 Defines if the facility "negotiation of the minimum throughput class" is added in the call packet (0 not added, 1 added). Default is 0.

–f5 Defines if the facility "reverse charging request" is added in the call packet (0 not added, 1 added). Default is 0.

–f6 Defines if the facility "fast select" is added in the call packet (0 no fast select, 1 fast select without restriction, 2 fast select with restriction). Default is 0.

–f7 Defines if the value of the transit delay is added in the call packet.

–f8 Defines if value of the end to end transit delay is added in the call packet.

–g5 Defines if value of the end to end transit delay is added in the call confirm packet.

–l List and displays all the defined PVC for the given Adaptercode

–r Remove the PVCs of PVCname of the given Adaptercode

–s Change/Show a defined PVC.

Examples

1. Define a PVC of name London–1, on the communications adapter 00–04–01–01, on the LCN 2345, for the Access method X25–3

   ```
   osix25 -C 00-04-01-01 London-1 2354 3610
   ```

2. Define 3 PVC on the X25 communications adapter 00–04, using the site name "PRIV". All the PVCs are managed by the transport.

   ```
   osix25 -c 00-04–01–01 -s PRIV -o1 0410 -o2 0410 -o3 0410
   ```

3. Add the negotiation of the packet size in the facilities of the call packets for the adapter 00–04–01–01.

   ```
   osix25 -f 00-04–01–01 -f1 1
   ```

4. List all PVC defined on the 00–04 adapter

   ```
   osix25 -l 00-04–01–01.
   ```

Implementation Specifics

Part of the OSI framework delivery.

Files

COMMANDS.cat	Contains the message catalog file.
/usr/bin/osix25	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration). Adapter location code.

Related Information

The **osiadapt** command, **osiconf** command, **osiload** command, **osiunload** command, **osinsap** command, **osirib** command, **osisnapshot** command, **OSIdefault** command, **osinetbios** command, **osiroute** command, **osirelay** command and **osifilter** command.

Chapter 11. OSI Stack Toolkit

This chapter explains how to exploit special management tools.

OSI Stack Toolkit: Overview

The tools described in this section supply the system administrator with management information not directly available through the System Management Interface Tool (SMIT).

You can find more information in:

- **cftran**: Provides TSAP management facilities for the TPAD–HPAD application, see page 11-2.
- **osiadapterinfo**: Provides adapter information facilities, see page 11-5.
- **osisas**: Defines Applications used by the SAS mechanism, see page 11-7.

Problem determination tools are provided using **ODIT**, see OSI Diagnostic Interactive Toolkit (ODIT) User's Guide, listed in the Bibliography, on page B-1.

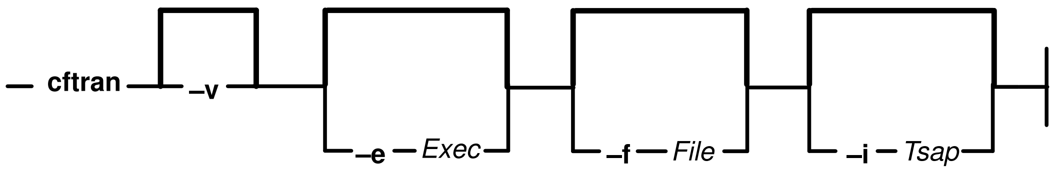
cftran Command (TSAP Management)

Purpose

The **cftran** command manages the Transport Service Access Point (TSAP) for the **TPAD–HPAD application only**. It allows the display and dynamic modification of the TSAP.

Syntax

cftran Command



Description

A menu drives the following functions:

- Addition of a TSAP
- Modification of a TSAP
- Deletion of a TSAP
- Display of all the TSAPs.

The command enables the use of the MAD tables in the Operating System memory image. Memory images can be used in real time from the UNIX kernel memory or off–line from a crash file.

This command must only to be used to add or remove TSAPs for the **TPAD–HPAD application** (i.e. TSAP associated with the module 0x46), or to list defined TSAPs.
Any other use is forbidden, but not controlled.

Modifications are lost at the next stack de–activation or next system reboot. If a persistent configuration is needed, write a script defining the TSAP and activate it after Stack activation (for example, using SAS), see **osisas** command, on page 11-7.

Parameters

–e *Exec* Executable file used to find the symbols of the AIX memory image (default value: */usr/lib/drivers/pse/osi/osi_low*).

–f *File* File to be modified (default value: */dev/kmem*).

–i *Tsap* TSAP number to be added, followed by the associated module unit (for example, –i 12ab34 4610).

–v Verbose mode (for non–interactive mode only).

Example

The command:

$ cftran

gives the following display:

```
************* Configuration of TSAP *************
exec file              : /etc/osistack
memory file            : /dev/kmem
|  No  |  Lg  |  Value of TSAP | Module | Unit |
|------|------|----------------|--------|------|
|  00  | 06   | 010203040506   |  46    |  10  |
|  01  | 06   | 060504030201   |  46    |  10  |
|  02  | 06   | 400150414431   |  46    |  10  |
|  03  | 06   | 400151373030   |  05    |  10  |
|  04  | 08   | 000000000000   |  05    |  10  |
|  05  | 04   | 00000000       |  05    |  10  |
|  06  | 06   | 400102030405   |  05    |  10  |
|  07  | 06   | 400105040302   |  05    |  10  |
|  08  | 06   | 40012a303031   |  09    |  00  |
|  09  | 06   | 400154523231   |  10    |  10  |
   1 : Add a TSAP          4 : List TSAPs
   2 : Modify a TSAP       x : Quit WITHOUT Save
   3 : Delete a TSAP       q : Save and Quit
  command =
*************************************************
```

The choice "command = **1**" gives the following sub-menu display:

```
*************************************************
  command = 1
 Value of TSAP : 400154523231
 Destination Module : 10
 Destination Unit : 10
|  0a  | 06   | 400154523231   |  10    |  10  |
  Type RETURN to Continue
```

```
************* Configuration of TSAP *************
exec file              : /etc/osistack
memory file            : /dev/kmem
|  No  |  Lg  |  Value of TSAP | Module | Unit |
|------|------|----------------|--------|------|
|  00  | 06   | 010203040506   |  46    |  10  |
|  01  | 06   | 060504030201   |  46    |  10  |
|  02  | 06   | 400150414431   |  46    |  10  |
|  03  | 06   | 400151373030   |  05    |  10  |
|  04  | 08   | 00000000000000 |  05    |  10  |
|  05  | 04   | 00000000       |  05    |  10  |
|  06  | 06   | 400102030405   |  05    |  10  |
|  07  | 06   | 400105040302   |  05    |  10  |
|  08  | 06   | 40012a303031   |  09    |  00  |
|  09  | 06   | 400154523231   |  10    |  10  |
|  0a  | 06   | 400154523231   |  10    |  10  |
   1 : Add a TSAP         4 : List TSAPs
   2 : Modify a TSAP      x : Quit WITHOUT Save
   3 : Delete a TSAP      q : Save and Quit
  command =
*************************************************
```

Usage

Meaning of symbols

No Chronological number assigned to the TSAP.

Lg Length of the TSAP (in bytes).

The other symbols are self explanatory.

osiadapterinfo (Adapter Information)

Purpose

The **osiadapterinfo** command provides the following information about all adapters and associated devices managed by the OSI Stack:

- Device name
- Adapter location
- Device state
- Medium Access Control (MAC) address or subscription.

Note: The device must be in the *Available* state before it can be used by the OSI driver.

Other possible states are:

Defined

Stopped.

In the *Available* state, the MAC address of the adapter is displayed for LAN adapters and the subscription is displayed for X.25 adapters.

Description

The name of the adapter is displayed in a verbose format, if the environment variables have been set to access catalogs. Depending on the platform on which the command is run, and on the adapters available in the machine, the names of the adapters can be:

- Ethernet High–Performance LAN Adapter
- DEC 2104X PCI Ethernet Controller
- Ethernet Standard Adapter
- Token Ring High–Performance Adapter
- FDDI Adapter (Fiber Single Ring, Fiber Dual Ring, UTP Single Ring). Use **lscfg –vl fddiX** command to get the exact type.
- FDDI Primary Card, Single Ring Fiber
- FDDI Primary Card, Dual Ring Fiber
- FDDI Primary Card, Single Ring Copper (STP)
- FDDI Primary Card, Dual Ring Copper (STP)
- X25 HiSpeed WAN Comm. Adapter

Note: The above list is subject to change.

Information about a single type of adapter is obtained by adding its unique type to the command line.

Syntax

osiadapterinfo Command

— **osiadapterinfo** ——— *UniqueType* ———|

Example

Typical results are shown below.

```
Ethernet High-Performance LAN Adapter
-------------------------------------
ent0    00-02   Available       02.60.8c.2c.a0.4c


Token-Ring High-Performance Adapter
-----------------------------------
tok0    00-04   Available       10.00.5a.a8.bd.3d
```

Suggested Reading

Prerequiste Information

Applications User documentation, listed in the *AIX and Related Products Documentation Overview,* see Bibliography, on page B-1.

Related Information

See OSI Applications & Communications Adapters Cross Reference in Appendix D.

osisas Command

Purpose

Manages the OSI Stack Application Support data base. It defines the application used by the SAS mechanism.

Syntax

osisas **–l**

osisas **–c** *Application [Application...]* **–s** *{ 'ON'|'OFF'}*

Description

The osisas command manages the enable / disable of the SAS mechanism onto specified applications. It supposes that the applications concerned have been declared to the SAS mechanism by adding the ISO_APPLICATIONS class object defined in the OSI framework delivery.

Note: If an OSI Stack is loaded, only the **list** option is available otherwise the command is rejected.

Parameters

Application	Name of the application to manage.

Flags

–l	Lists the applications declared to the SAS mechanism and their state.
–c	Changes the state of the applications named **Application**.
–s	Specifies the new state.

Examples

1. osisas –l

```
# Application              State
ftam                       OFF
uft                        OFF
```

2. osisas –c ftam uft –s OFF

```
Do Not Start ftam on osiload
Do Not Start uft on osiload
```

Implementation Specifics

Part of the OSI framework delivery.

Files

OSICONF.cat	Contains the message catalog file.
usr/bin/osisas	Specifies the command file.

Suggested Reading

Prerequisite Information

OSI services Reference Manual (OSI communications stack, OSI Stack configuration).

Related Information

The **osiload** command and **osiunload** command.

Appendix A. OSI Telecommunications Profiles

This appendix describes the profiles available to applications using the OSI services.

OSI Telecommunications Profiles Overview

This overview contains details of the profiles supported for applications using the OSI services. Correspondence with standardized profiles is provided, when applicable. You can find more information in:

- LAN/1 on page A-2
- LAN/2 on page A-2
- LAN/3 on page A-2
- LAN/4 on page A-3
- LAN/6 on page A-3
- X.25/1 on page A-3
- X.25/2 on page A-4
- X.25/5 on page A-5
- REL/1bis on page A-5
- REL/2 on page A-5
- REL/2bis on page A-6
- REL/2ter on page A-6
- SID Profiles on page A-6

LAN/1

COTS over Null Internet on CSMA/CD

ISO 8073/ADD 2
ISO 8473 Inactive
ISO 8802.2 LLC 1
ISO 8802.3

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/6211	CEN–CENELEC	ENV 41101	Stable	Inactive Subset

LAN/2

COTS over Full Internet on CSMA/CD

ISO 8073/ADD 2
ISO 8473 Full
ISO 8802.2 LLC 1
ISO 8802.3

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/A51	CEN–CENELEC	ENV 41102	Stable	Full Internet Opt: Inactive
T/A51–	NIST	NIST SP500–183 Parts 2, 3 & 4	Stable	Full Internet
T/A51–	ISO	ISP TA 51	Stable	Full Internet

LAN/3

COTS over Null Internet on Token Ring

ISO 8073/ADD 2
ISO 8473 Inactive
ISO 8802.2 LLC 1
ISO 8802.5

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/6231	CEN–CENELEC	ENV 41109	Stable	Inactive Subset

LAN/4

COTS over Full Internet on Token Ring

ISO 8073/ADD 2
ISO 8473 Full
ISO 8802.2 LLC 1
ISO 8802.5

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/A53	CEN–CENELEC	prENV 41110	Stable	Full Internet
T/A53–	NIST	NIST SP500–183 Parts 2, 3 & 4	Stable	Full Internet
T/A53–	ISO	pdISP TA 53	Unstable	Full Internet

LAN/6

COTS over Full Internet on FDDI

ISO 8073/ADD 2
ISO 8473 Full
ISO 8802.2 LLC 1
ISO 9314

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/A54	NIST	NIST SP500–183 Parts 2, 3 & 4	Stable	Full Internet
T/A54	ISO	pdISP TA 54	Unstable	Full Internet

Note: No profile has been identified for Null Internet over FDDI, but this feature is supported by the Stack.

X.25/1

COTS over X.25 permanent access through telephone network or X.21 network.

ISO 8073
ISO 8208 DTE to DCE
ISO 7776
X.21 bis

This profile is to be used for mapping transport directly on to X.25, to access PSDNs over a leased line.

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/x11x1	CEN–CENELEC	ENV 41101 Pt2	Unstable	Cl. 0, or 0 and 2
	NIST	NIST SP 500–183 Parts 2, 3 & 4	Stable	Cl. 0, 2 and 4
–	ISO	ISP TB 1111	Stable	Cl. 0, 2 and 4 (VC)/PSTN
–	ISO	ISP TC 1111	Stable	Cl. 0 and 2 (VC)/PSTN
–	ISO	ISP TD 1111	Stable	Cl. 0 (VC)/PSTN
–	ISO	ISP TE 1111	Stable	Cl.2 (VC)/PSTN
–	ISO	ISP TB 1121	Stable	Cl. 0, 2 and 4 (VC)/CSDN
–	ISO	ISP TC 1121	Stable	Cl. 0 and 2 (VC)/CSDN
–	ISO	ISP TD 1121	Stable	Cl. 0 (VC)/CSDN
–	ISO	ISP TE 1121	Stable	Cl. 2 (VC)/CSTN

X.25/2

COTS over CLNS mapped on X.25 permanent access through a Public Switched Telephone Network (PSTN) or a Circuit Switched Data Network (CSDN)

ISO 8073/ADD 2
ISO 8473 ISO 8208
ISO 7776
X.21 bis

This profile uses SNDCF for operating ISO 8473 over ISO 8208. It is to be used for access to PSDNs through CLNS over a leased line. The leased line on a public switched telephone network or circuit switched data network is the permanent access to the PSDN.

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
TA/11n1	NIST	NIST SP500–183 Parts 2, 3 & 4	Stable	CLNS over X.25
TA/11n1	ISO	ISP TA 1111	Stable	CLNS over X.25
TA/11n1	ISO	ISP TA 1121	Stable	CLNS over X.25

X.25/5

COTS over X.25 switched access through PSTN or CSDN.

ISO 8073
ISO 8208 DTE to DCE
ISO 7776
X.21 or X.21 bis

This profile requires that the X.25 low layers are able to establish the access to the CSDN or PSTN by numbering. It must be performed with external numbering machines.

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
T/3221	CEN–CENELEC	ENV 41105 pt2	Stable	PSTN Cl. 0 and 2
T/3222	CEN–CENELEC	ENV 41105 pt2	Stable	CSDN Cl. 0 and 2

REL/1bis

This profile is based on ISO TR 10172, applying the MSDSG technique.

ISO 8073 ADD 2	ISO 8073
ISO 8473	ISO 8208
ISO 8802.2	ISO 7776
ISO 8802.3	X.21 or X.21 bis

REL/2

Relaying the CLNS LAN CSMA/CD to LAN CSMA/CD

ISO 8473	
ISO 8802.2	ISO 8802.2
ISO 8802.3	ISO 8802.3

This profile includes the use of ISO 9542, 88 for intermediate systems.

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
R/A51.51	CEN–CENELEC	ENV 41801	Stable	–
R/A51.51	ISO	pdISP RA51.51	Unstable	–

REL/2bis

Relaying the CLNS LAN CSMA/CD to PSDN with mapping of Internet on X.25.

<table>
<tr><td rowspan="2">ISO 8473</td><td>ISO 8473</td></tr>
<tr><td>ISO 8208</td></tr>
<tr><td>ISO 8802.2</td><td>ISO 7776</td></tr>
<tr><td>ISO 8802.3</td><td>X.21 or X.21 bis</td></tr>
</table>

Note: This profile uses SNDCF for operating ISO 8473 over ISO 8208.

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
R/A51.1111	CEN–CENELEC	ENV 41801	Unstable	PSTN
R/A51.1121	CEN–CENELEC	ENV 41801	Unstable	CSDN
R/A51.1111	ISO	pdISP RA 51.1111	Unstable	PSTN
R/A51.1121	ISO	pdISP RA 51.1121	Unstable	CSDN

REL/2ter

Relaying the CLNS LAN CSMA/CD to LAN FDDI

<table>
<tr><td colspan="2">ISO 8473</td></tr>
<tr><td>ISO 8802.2</td><td>ISO 8802.2</td></tr>
<tr><td>ISO 8802.3</td><td>ISO 8802.3</td></tr>
</table>

Corresponding Functional Standards:

CEN–ID	Organization	Reference	Status	Comments
R/A51.54	ISO	RA51.54	On–going	–

SID Profiles

The SID profile for use over the10 Base 5 CSMA/CD LAN is:

Reference	Status	Comments
SID1.00, 15 A(F)2 9925 Rev1 Sections 3.5 & 5	Stable	–

This profile uses the inactive subset of Internet (ISO 8473) and the support of ES/IS (ISO 9542) is not applicable in conjunction with that profile.

Note: The profile uses a DSA specific LSAP selector and thus does not conform to ISO addressing rules. The network addresses are defined with the SNPAs (Ethernet address).

Appendix B. Bibliography

This appendix gives information about the document references used in this guide.

This overview provides information about applications documentation, lists of related documentation, standards and License Control information. You can find more information in:

List of Applications Documentation

Applications documentation is as follows:

NetShare

NetShare User's Guide	86 A2 95AP

Other applications documentation is listed in the *AIX and Related Products Documentation Overview*.

List of Related Documents (Referenced in Text)

Related documents referenced in the text, are listed in the following table:

BASIC-COM-MP and GCOS-COM-MP Installation Guide	86 A2 74AT
AIX and Related Products Documentation Overview	86 A2 71WE
Site Preparation Guide	86 A1 06WD
HiSpeed WAN Comm. Installation and Service Guide	86 A1 81WG
OSI Diagnostic Interactive Toolkit (ODIT) User's Guide	86 A2 39WG
OSI Communications Porting Guide	86 A2 44AP
AIX Installation Guide	86 A2 60AP
MAX25-3 API Programmer's Guide	86 A2 45AP
Connection Information Service (CIS) Installation, Administration & User's Guide	86 A2 26AP
Adapters, Devices, and Cable Information	86 A1 76AT

License Control

License Control details are provided in the following documents:

iFOR/LS System Management Guide	86 A2 11AQ
iFOR/LS Tips & Techniques	86 A2 12AQ

Standards

This section provides a list of standards. You can find more information in:

- ISO Standards on page B-2
- CCITT Recommendations on page B-4
- X/Open Guides on page B-4
- International Standardized Profiles (ISP) on page B-4
- SPAG Profiles on page B-5
- CEN/CENELEC Profiles on page B-5
- Streams Interfaces on page B-5
- RFC on page B-5
- Government Profiles on page B-5

ISO Standards

The ISO standards are listed in the following table:

[ISO 7498–1, 91]	OSI Basic Reference Model
[ISO 7498–2]	OSI Reference Model – Security Architecture
[ISO 7498–3, 88]	OSI Reference Model – Naming and Addressing
[ISO 7776, 86]	High–level data link control procedures – Description of the X25 LAP–B compatible DTE link procedures
[ISO 7809]	Consolidation of classes of procedures
[ISO 8072, 86]	Transport service definition
[ISO 8073, 88]	Connection–oriented transport protocol definition
[ISO 8208, 90]	X25 packet level protocol for Data Terminal Equipment
[ISO 8326 DAD2, 88]	Basic Connection–oriented session service definition– Addendum 2: Incorporation of unlimited User Data.
[ISO 8326 DAM4, 91]	Basic Connection–Oriented Session Service Definition– Amendment 4: Additional Synchronization Functionality.
[ISO 8326, 87]	Basic Connection–oriented session service definition
[ISO 8327 DAD2, 88]	Basic Connection–Oriented Session Protocol Specification – Addendum 2: Incorporation of unlimited User Data.
[ISO 8327 DAM3, 91]	Basic Connection–Oriented Session Protocol Specification – Amendment 3: Additional Synchronization Functionality.
[ISO 8327, 87]	Basic Connection–Oriented Session Protocol Specification
[ISO 8348/AD1]	Addendum to ISO 8348 covering Connectionless Transmission Mode
[ISO 8348/AD2]	Addendum to ISO 8348 covering Network layer Addressing
[ISO 8348/AD3]	Addendum to ISO 8348 covering Additional features for the network service
[ISO 8348]	Network service definition
[ISO 8473, 88]	Protocol for providing the connectionless–mode network service
[ISO 8648]	Internal organization of the network layer
[ISO 8649, 88]	Service definition for the Association Control Service Element (ACSE)
[ISO 8650 TC1]	Protocol specification for the Association Control Service Element (ACSE) – Amendment 1: authentification.

[ISO 8650, 88]	Protocol specification for the Association Control Service Element (ACSE)
[ISO 8802–1, 89]	Local Area Networks Protocols
[ISO 8802–2, 89]	Local Area Networks – Logical Link Control (LLC)
[ISO 8802–3]	Local Area Networks – CSMA/CD access method and physical layer specification
[ISO 8802–5]	Local Area Networks – Token Ring access method and physical layer specification
[ISO 8822, 88]	Connection–oriented presentation service definition
[ISO 8823, 88]	Connection–oriented presentation protocol definition
[ISO 8824]	Specification of Abstract Syntax One (ASN.1)
[ISO 8825]	Specification of Basic Encoding Rules for Abstract Syntax One (ASN.1)
[ISO 8878, 87]	use of X.25 to provide the OSI connection oriented network service
[ISO 8878–2, 91]	Use of X25 to provide the OSI connection–mode network service – Part 2: PICS
[ISO 8878/AD1, 90]	use of X.25 to provide the OSI connection oriented network service / Addendum 1: priority
[ISO 8878/AD2, 90]	Use of X25 to provide the OSI connection–mode network service / Addendum 2: use of PVC to provide the OSI CONS
[ISO 8880–1]	Specification of protocols to provide and support the OSI network service – Part one : general principles
[ISO 8880–2]	Specification of protocols to provide and support the OSI network service – Part two : provision and support of the connection–mode network service (CONS)
[ISO 8880–3]	Specification of protocols to provide and support the OSI network service – Part three : precision and support of the connectionless–mode network service (CLNS)
[ISO 9072–1, 89]	Remote Operations Part 1: ROSE model, notation and service definition
[ISO 9072–2, 89]	Remote Operations Part 2: ROSE protocol definition
[ISO 9314–1]	FDDI – Physical layer
[ISO 9314–2]	FDDI – Medium Access Control
[ISO 9314–3]	Medium dependent Physical Layer
[ISO 9542, 88]	End System to Intermediate System routing exchange protocol for use in conjunction with ISO 8473
[ISO 10030, 90]	End System Routing Information Exchange Protocol for use in conjunction with ISO 8878.
[ISO 10589, 92]	Intermediate System to Intermediate System Intra Domain routing information exchange protocol for use in conjunction with the protocol for providing the Connectionless–mode Network Service
[ISO TR 10172, 91]	Network / Transport Protocol Interworking Specification
[ISO TR 10172]	CO/CR Interworking Functional Units
[ISO TR 10178]	The structure and coding of link service access point addresses in LANs
[ISO TR 8509]	OSI Layer Service Definition Conventions
[ISO TR 9575, 89]	OSI Routing framework

CCITT Recommendations

The CCITT Recommendations are listed in the following table:

[CCITT X21]	Interface between DTE and DCE for synchronous operation on public data networks
[CCITT X25, 88]	Interface between Data Terminal Equipment (DTE) and Data Circuit–terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit (CCITT *Blue Book*)
[CCITT I440]	LAP–D
[CCITT I441]	LAP–D

X/Open Guides

The X/Open guides are listed in the following table:

[XTI, 92]	X/Open Transport Interface – CAE Specification – XPG4 – January 1992
[XAP, 93]	X/Open ACSE/Presentation Services Application Programming Interface – CAE Specification – May September 1993
[XTIX25, 93]	Appendix: use of XTI to support X.25 – V 0.8, October 1993

International Standardized Profiles (ISP)

The International Standardized profiles are listed in the following table:

[ISO TR 10000–1]	Taxonomy Framework
[ISO TR 10000–2]	Taxonomy of Profiles
[ISP 10608–1]	Provision of the COTS over the CLNS – Subnetwork Independent Requirements
[ISP 10608–2]	Provision of the COTS over the CLNS – Definition of profile TA51
[ISP 10608–4]	Provision of the COTS over the CLNS – Definition of profile TA53
[ISP 10608–5]	Provision of the COTS over the CLNS – Definition of profiles TA 1111/1121
[ISP 10608–6]	Provision of the COTS over the CLNS – Definition of profile TA 54 for operation over an FDDI LAN subnetwork (1992)
[ISP 10608–14]	Provision of the COTS over the CLNS – MAC, PHY and PMD Sublayer dependent and Station Management requirements over an FDDI LAN subnetwork (1992)
[ISP 10609–1]	Provision of the COTS over the CONS – Group TB requirements
[ISP 10609–5]	Provision of the COTS over the CONS – Definition of profiles TB 1111/1121

SPAG Profiles

The SPAG profiles are listed in the following table:

[T31]	Permanent Access to a Packet Switched Data Network
[T32]	Switched Access to a Packet Switched Data Network
[R21]	Connectionless Internet Relay (LAN–LAN)
[R22]	Connectionless Internet Relay (LAN–PSDN–LAN)
[T6211]	CSMA/CD LAN (Null Internet)
[T6212]	CSMA/CD LAN (Connectionless–mode Internet)

CEN/CENELEC Profiles

The CEN/CENELEC profiles are listed in the following table:

[ENV 41101]	Provision of the COTS over the CLNS using a CSMA/CD LAN isolated
[ENV 41102]	Provision of the COTS over the CLNS using a CSMA/CD LAN
[ENV 41104/2]	Provision of the COTS through the CONS through a PSDN, permanent access case
[ENV 41105/2]	Provision of the COTS over the CONS through a PSDN, switched access case
[ENV 41107]	Provision of the COTS over the CONS through a CSDN
[ENV 41109]	Provision of the COTS over the CLNS using a Token Ring LAN isolated
[ENV 41110]	Provision of the COTS over the CLNS using a Token Ring LAN
[ENV 41801]	Provision of the network relay function in a relay system attached to two or more LANs to allow interworking among End systems providing the OSI CLNS. (1992)

Streams Interfaces

The STREAMS Interfaces are listed in the following table:

[TPI, 92]	UNIX System V Porting Rules – A Streams based Transport Provider Interface– AT&T Bell Laboratories – version 1.5

Request For Comments

The Request For Comments (RFCs) are listed in the following table:

[RFC1006, 87]	OSI Transport Service on top of the TCP, version 3
[RFC1277, 91]	Encoding Network Address to support operation over non OSI lower layers

Government Open System Profiles

The Government Open System Profiles are listed in the following table:

[UKGOSIP]	GOSIP 4 UK Government OSI Profile. Fourth Edition
[USGOSIP2]	Government Open Systems Interconnection Profile. FIPS PUB 146–1. 1991 April 3
[SIA]	Stable Implementation Agreements for Open System Interconnection protocol. Version 4 Edition 1. December 1990.

Appendix C. OSI Configuration Error Codes & Messages

This appendix describes the error codes and messages that can be returned by the system.

Overview

The error codes and messages returned by the system can be divided into two parts. You can find more information in:

- OSI Stack Configurator on page C-2
- OSI Start Applications Support on page C-25

Error codes and messages concerning the particular telecommunication functions are not listed here. Their details are provided in the products' respective programmers guides and reference manuals. These products are introduced in Introducing OSI Communications on page 1-1. The relevant documents are listed in the *AIX and Related Products Documentation Overview,* see Bibliography, on page B-1.

For OSI Stack problem determination, refer to *OSI Diagnostic Interactive Toolkit (ODIT) User's Guide,* see Bibliography, on page B-1.

OSI Stack Configurator

This section shows the messages displayed when using the System Management Interface Tool (SMIT) to manage the OSI Stack.

You can find more information under the headings:

- Add a local NSAP on page C-8.
- Add a Profile entry from a model on page C-12.
- Add a PVC on page C-16.
- Add a Relay route entry from a model on page C-11.
- Add a Route entry on page C-13.
- Add a static RIB entry from a model on page C-9.
- Add communications adapters on page C-6.
- Change configuration name or comment on page C-5.
- Change/show a PVC on page C-16.
- Change/Show a Communications adapter on page C-7.
- Change/show a static RIB entry on page C-10.
- COSP on page C-18.
- COTP on page C-19.
- Create a configuration on page C-5.
- Define Netbios configuration on page C-21.
- Display ES–IS status on page C-24.
- ES–IS on page C-19.
- List all configurations on page C-5.
- List communications adapters on page C-6.
- List defined profile entries on page C-12.
- List defined route entries on page C-13.
- List defined Relay route entries on page C-11.
- List dynamic RIB on page C-24.
- List local NSAPs on page C-7.
- List static RIB on page C-9.
- List the PVCs on page C-16.
- List wait entries on page C-24.
- Load of last selected configuration on page C-23.
- Memory management on page C-20.
- Minimum Configuration on page C-4.
- OSI Layers configuration on page C-14.

Minimum Configuration

The Minimum Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
Minimum configuration See also: Create a configuration Add communications adapters Add a local NSAP Show a configuration	– The OSI default configuration is already defined.	– <All the parameters of the created configuration>

Configuration Management

The Configuration Management menu messages are shown in the following table:

Key Word	Errors	Responses
List all configurations	– Configuration list is empty.	– <The list of the defined configurations with: name, date, comment and state>
Create a configuration	– Permission denied: command for super user only. – A configuration has already the <conf–name> name, not created. – Selected configuration not found.	The configuration <conf–name> has been created.
Remove a configuration	– Permission denied: command for super user only. – This configuration is loaded; Remove not allowed. – Configuration list is empty. – Configuration <conf–name> not found.	– The configuration <conf–name> has been removed.
Select a configuration	Configuration list is empty. Configuration <conf–name> not found. This configuration was already selected.	– The configuration <conf–name> has been selected.
Show a configuration	– Configuration list is empty. – Configuration <conf–name> not found.	– <All the parameters of a configuration>
Change configuration name or comment	– Permission denied: command for super user only. – Configuration list is empty. – Configuration <conf–name> not found. – A configuration has already the <conf–name> name, not changed. – This configuration is loaded; Rename not allowed.	– The name of the configuration <old–conf–name> has been modified to <new–conf–name>. – The comment of the configuration <conf–name> has been modified.

Configuration Definition

You can find more information on the configuration definition menu messages in:

- Communications Adapter Access on page C-6
- Addressing Management on page C-7

Communications Adapter Access

The Communications Adapter Access menu messages are shown in the following table:

Key Word	Errors	Responses
	General errors: – A configuration must have been selected before. – Erroneous adapter code <adapter–code>, format is AA–BB [–CC [–DD [–EE]]].	
List communications adapters	– No adapter defined in this configuration. – No adapter defined in this configuration.	– < The list of the communications adapters with: network–type and adapter–code.>
Add communications adapters	– Permission denied: command for super user only. – Not more than <number> <adapter–type> can be defined. – The communications adapter <adapter–code> <adapter–type> is already defined. – Cannot Add: Communications adapter has already the <adapter–code> code with another type. – The adapter <adapter–code> is not Available in the machine. – Bad Ethernet Maximum NPDU: <max–npdu> – Bad Token Ring Maximum NPDU: <max–npdu> – Bad FDDI Maximum NPDU: <max–npdu> – Cannot attach adapter <adapter–code> <adapter–type>. – Cannot start dat_x25 daemon. Cannot attach subdomain for adapter <adapter–code> < adapter–type> .	– The communications adapter <adapter–code> <adapter–type> has been configured. – The communications adapter <adapter–code> <adapter–type> has been attached.

Key Word	Errors	Responses
Change/Show a communications adapter	– Permission denied: command for super user only.	– The communications adapter <adapter–code > <adapter–type> has been modified.
	– Cannot modify: the new Adapter code <adapter–code>is already defined.	– The communications adapter has been attached.
	– Cannot modify: the new Adapter code <adapter–code>is not of the same type.	
	– Cannot modify: another communicatons adapter has already the <subdomain> subdomain with up state.	
	– The adapter <adapter–code> is not Available in the machine.	
	– Bad Ethernet Maximum NPDU: <max–npdu>	
	– Bad Token Ring Maximum NPDU: <max–npdu>	
	– Bad FDDI Maximum NPDU: <max–npdu>	
	– Cannot attach adapter <adapter–code> <adapter–type>.	
	Cannot attach subdomain for adapter <adapter–code > <adapter–type>.	
	– Cannot start dat_x25 daemon.	
Remove communications adapters	– Permission denied: command for super user only.	– The communications adapter <adapter–code> has been removed.
	– The adapter <adapter–code> has not been found.	– The X25 facilities of the adapter <adapter–code> have been removed.
	– Cannot detach adapter <adapter–type> <adapter–code>.	– The PVCs of the adapter <adapter–code> have been removed.
		– The RIB entries of the adapter <adapter–code> have been removed.

Addressing Management

The Addressing Management menu messages are shown in the following table:

Key Word	Errors	Responses
	General Errors:	
	– A configuration must have been selected before.	
	– Permission denied: Lower layers not installed.	
List local NSAPs	– No local NSAP defined in this configuration.	– <List of the local NSAPs>

Key Word	Errors	Responses
Add a local NSAP	– Permission denied: command for super user only. – This local NSAP has already been defined. Not more than <number> local NSAPs can be defined. – Update in Loaded Configuration: Error, no local NSAP added.	– Update in Loaded Configuration: Local NSAP <nsap> has been added in memory. – The local NSAP <nsap> has been added.
Remove local NSAPs	– Permission denied: command for super user only. – The local NSAP <nsap> is not defined. – Local NSAP is invalid. – No local NSAP defined.	– The local NSAP <nsap> has been removed.

RIB Entries Management

The RIB Entries Management menu messages are shown in the following table:

Key Word	Errors	Responses
List static RIB	– No RIB entries defined in this configuration.	– < The list of the RIB entries with all attributes for each entry.>
Add a Static RIB entry from a model	– Permission denied: command for super user only. – The network type must be in accordance with the subnet entry. – A RIB entry with the same NSAP, Mask, Subdomain and remote SNPA already exists. – The RIB entry is not valid. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. Static RIB not reset. – Bad Subdomain: <subdomain> – Bad NSAP: <nsap> – Bad internal number: <int–num> – Bad RIB type: <type> – Bad address: <address> – Bad NSAP mask: <mask> – Bad NSAP priority: <priority> – Bad life time: <lifetime> – Bad Closed User Group number: <CUG> – Bad RIB state: <state> – Bad RIB part: <part> – Bad NSAP type: <type> – Bad SNPA type: <type> – Bad SNPA name: <snpa> – The STATICF RIB entries part is full. – NOTE: The RIBpart parameter of this entry is forced to STATICF.	– The RIB entry on the NSAP %s has been defined. – This configuration is loaded. Static RIB reset.

Key Word	Errors	Responses
Change/show a Static RIB entry	– Permission denied: command for super user only. – The RIB entry is not defined. (See other errors in *Add a Static RIB entry from a model)*	– The RIB entry has been changed. – This configuration is loaded. Static RIB reset.
Remove Static RIB entries	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. Static RIB not reset.	– The RIB entry with internal identifier <ident> has been removed. – This configuration is loaded. Static RIB reset.

Relay Route Entries Management

The Relay Route Entries Management menu messages are shown in the following table:

Key Word	Errors	Responses
List defined Relay route entries	– No Relays routes have been defined in this configuration.	– <The list of the Relay route entries with all attributes for each entry>
Add a Relay route entry from a model	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. **WARNING:** The loaded Relays routes table cannot be cleared. – This configuration is loaded. **WARNING:** Loading Relays routes table FAILED. – This configuration is loaded. The Relays routes table cannot be loaded. – Bad Subdomain: <value> – Bad TSEL: <value> – Bad Address: <value> – Bad Transport class: <value> – Bad Address type: <value> – The <RelayName> Relay route Name is already defined. – The <RelayName> Relay route has the same key than the Relay route <RelayName>.	– <The <RelayName> Relay route has been added. – This configuration is loaded. The loaded Relays routes table has been cleared. The loaded Relays routes table has been loaded.>
Remove Relay route entries	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. **WARNING:** The loaded Relays routes table cannot be cleared. – This configuration is loaded. **WARNING:** Loading Relays routes table FAILED. – This configuration is loaded. The Relays routes table cannot be loaded.	– <The <RelayName> Relay route is not defined. – The <RelayName> Relay route has been removed. – This configuration is loaded. The loaded Relays routes table has been cleared. The Relays routes table has been loaded.>

Profile Entries Management

The Profile Entries Management menu messages are shown in the following table:

Key Word	Errors	Responses
List defined profile entries	– No profiles have been defined in this configuration.	– <The list of the profile entries with all attributes for each entry>
Add a Profile entry from a model	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. **WARNING:** The loaded Profiles table cannot be cleared. – This configuration is loaded. **WARNING:** Loading Profile table FAILED. – This configuration is loaded. The Profiles table cannot be loaded. – Bad Transport class: <value> – Bad Alternate Transport class: <value> – Bad flag value: <value> – Bad UseExpData value: <value> – Bad UseFlowCtrl value: <value> – Bad UseExtendFmt value: <value> – Bad UseChecksum value: <value> – Bad Priority: <value> – Bad Credit: <value> – Bad TPDUsize: <value> – The <ProfileName> Profile is already defined.	– < The <ProfileName> Profile has been added. – This configuration is loaded. The loaded Profiles table has been cleared. The Profiles table has been loaded.>
Remove Profile entries	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. WARNING: The loaded Profiles table cannot be cleared. – This configuration is loaded. WARNING: Loading Profile table FAILED. – This configuration is loaded. The Profiles table cannot be loaded. – The <ProfileName> Profile is not defined.	– < <number> Routes have been removed. – The <ProfileName> Profile has been removed. – This configuration is loaded. The loaded Profiles table has been cleared. The Profiles table has been loaded.>

Route Entries Management

The Route Entries Management menu messages are shown in the following table:

Key Word	Errors	Responses
List defined route entries	– No routes have been defined in this configuration. – No Routes have been defined for the Profile <ProfileName>.	– <The list of the route entries with all attributes for each entry>
Add a Route entry	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction. – This configuration is loaded. Static RIB not reset. – This configuration is loaded. **WARNING:** The loaded Routes table cannot be cleared. – This configuration is loaded. **WARNING:** Loading Route table FAILED. – This configuration is loaded. The Routes table cannot be loaded. – The <RouteName> Route is already defined. – This <RouteName> Route has the same Type and Value as the <RouteName> Route. – The Profile Tclass <value> is wrong for the RouteType <value>– Bad RouteValue: <value> – Bad RouteType: <value>	– <The <RouteName> Route has been added. – This configuration is loaded. The loaded Routes table has been cleared. The Routes table has been loaded.>
Remove Route entries	– Permission denied: command for super user only. – This configuration is loaded. Stack is not active. – This configuration is loaded. Error in MAD transaction.	– <The <RouteName> Route has been removed.– This configuration is loaded. The loaded Routes table has been cleared. The Routes table has been loaded.>

OSI Layers Configuration

The OSI Layers Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
OSI Layers Configuration	– Permission denied: command for super user only. – A configuration must have been selected before. – Permission denied: upper layers not installed. – Permission denied: lower layers not installed. – Configuration <conf–name> not found. – Bad no. of COPP connections: <number> – Bad no. of COSP connections: <number> – Bad no. of COTP connections: <number> – Bad multiplexing rate: <number> – The max. no. of COPP connections must be in the range from 1 to 1024. – The max. no. of COSP connections must be in the range from 1 to 1024. – The max. no. of COTP connections must be in the range from 1 to 1024. – The max. no. of COTP connections multiplexed on a VC must be in the range from 1 to 1024. – The max. no. of COPP connections must be lower than the max. no. of COSP connections. – The max. no. of COTP connections multiplexed on a VC must be lower than the max. no. of COTP connections. – Configuration name: <conf–name>. This configuration is loaded. The no. of COTP connections on a VC cannot be modified. – Configuration name: <conf–name>. This configuration is loaded. The type of ES–IS function cannot be modified. – Update in loaded configuration: The max. no. of COPP connections cannot be modified.	– OSI configuration has been modified. – Update in loaded configuration: The max. no. of COPP connections has been modified. – Update in loaded configuration: The max. no. of COSP connections has been modified. – Update in loaded configuration: The max. no. of COTP connections has been modified. – Update in loaded configuration: The activation of the ES–IS protocol has been modified. ES–IS is <ENABLE or DISABLE> – The selected configuration has been modified.

Key Word	Errors	Responses
	– Update in loaded configuration: The max. no. of COSP connections cannot be modified.	
	– Update in loaded configuration: The max. no. of COTP connections cannot be modified.	
	– Update in loaded configuration: The activation of the ES–IS protocol cannot be modified.	
	– Update in loaded configuration: An error occurred when updating in memory. Data base not modified.	

X25 Configuration

The X25 Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
	General Errors: – A configuration must have been selected before. – Permission denied: lower layers not installed.	

Permanent Virtual Circuit Management

The PVC management menu messages are shown in the following table:

Key Word	Errors	Responses
List the PVCs	– No X25 type communications adapters are defined. – No PVCs are defined.	– <The list of defined PVCs with: name, adapter–code, channel and owner>.
Add a PVC	– Permission denied: command for super user only. – No X25 type communications adapters are defined. – The PVC name <name> is already defined for another adapter. – The Logical Channel number <channel> is already defined. – The PVC is already defined. – Bad Channel value <value> (Allowed: [0..255]). – Bad Owner value <value> (Allowed: 0000, 0410, 3610). – Bad Channel value <value> (Allowed: [0..4095]). – The adapter <adapter–code> is not an X25 adapter. – The adapter <adapter–code> is not defined in the configuration. – This configuration is loaded. No PVC can be added, modified or removed. – All available PVCs for this <adapter–code> adapter have been defined. – A CuDv ODM object is missing for this <adapter–code> adapter.	– The X25 PVC <name> on the adapter <adapter–code> has been added.

Change/Show a PVC	– Permission denied: command for super user only. – No X25 type communications adapters are defined. – The PVC <name> does not exist. – No PVCs are defined. (See also *Add a PVC*)	– The X25 PVC <name> on adapter <adapter–code> has been modified.
Remove a PVC	– Permission denied: command for super user only. – No X25 type communications adapters are defined. – This configuration is loaded. No PVC can be added, modified or removed.	– The PVC <name> has been removed.

Subsequent Protocol Identifier (SPI) Table

The SPI table menu messages are shown in the following table:

Key Word	Errors	Responses
	– Permission denied: command for super user only.	– The Subsequent Protocol Identifier Table has been modified.

X25 Facilities

The SPI table menu messages are shown in the following table:

Key Word	Errors	Responses
	– Permission denied: command for super user only. – No X25 type communications adapters are defined. – The value of the transit delay must be in the range from 1 to 65535. – The value of the end to end transit delay must be in the range from 1 to 65535. – This configuration is loaded. The X25 facilities cannot be modified. – The adapter <adapter–code> is not an X25 adapter. – The adapter <adapter–code> is not defined in the configurator. – The X25 subscription <adapter–code> has not been modified.	– The X25 facilities have been defined. – The X25 facilities have been modified.

Tuning Configuration

The tuning configuration menu messages are shown in the following table:

Key Word	Errors	Responses
	General Errors: – A configuration must have been selected before.	
Connection Oriented Session Protocol	– Permission denied: command for super user only. – Permission denied: upper layers not installed. – Bad COSP connection timer: <value> – Bad COSP disconnection timer: <value> – Bad COSP indication connection lifetime: <value> – The COSP connection timer value must be in the range from 160 to 20000 seconds. – The COSP disconnection timer value must be in the range from 40 to 20000 seconds. – The COSP indication connection lifetime value must be in the range from 1 to 20000 seconds. – This configuration is loaded. The COSP indication connection lifetime cannot be modified. – Update in Loaded Configuration: The COSP connection timer cannot be modified. – Update in Loaded Configuration: The COSP disconnection timer cannot be modified. – This configuration is loaded. An error occurred when updating in memory, Data base not modified.	– Update in Loaded Configuration: The COSP connection timer has been modified. – Update in Loaded Configuration: The COSP disconnection timer has been modified. – COSP timers modified.

Connection Oriented Transport Protocol	– Permission denied: command for super user only. – Permission denied: lower layers not installed. – Bad COTP timer: <value> – Bad no. of COTP retransmissions: <value> – The COTP local retransmission time value must be in the range from 500 to 20000 milliseconds. – The max. no. of transmissions of a COTP PDU value (N) must be in the range from 1 to 15. – Update in Loaded Configuration: The COTP timer cannot be modified. – Update in Loaded Configuration: The COTP retransmissions cannot be modified. – This configuration is loaded. An error occurred when updating in memory, Data base not modified.	– Update in Loaded Configuration: The COTP timer has been modified. – Update in Loaded Configuration: The COTP retransmission has been modified. – COTP timer modified. – COTP retransmissions modified.
End System – Intermediate System (ES–IS)	– Permission denied: command for super user only. – Permission denied: lower layers not installed. – Bad ES timer: <value> – Bad IS timer: <value> – The value of the ES timer must be in the range from 1 to 20000 seconds. – The value of the IS timer must be in the range from 1 to 20000 seconds. – This configuration is loaded. ES or IS timer cannot be modified.	– ES–IS timers modified.

Memory Management	– Permission denied: command for super user only. – The memory size value must be in the range from 1000 to 2000000 kbytes. – This configuration is loaded. The max. memory size cannot be modified. – This configuration is loaded. An error occurred when updating in memory, Data base not modified.	– This configuration is loaded. The max. memory size has been modified. – Buffer memory size modified.
NULL CLNP profile network LSAPs definition	– Permission denied: command for super user only. – This configuration is loaded. Command not allowed: the Configuration needs to be unloaded. – The Inactive Selectors (LSAPs) cannot be defined. – Wrong parameter: <value>	– The Inactive Selectors (LSAPs) have been defined.

Netbios Configuration

The Netbios Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
Show Netbios configuration	–	– <The defined (or default) Netbios attributes.>
Define Netbios configuration	– Permission denied: command for super user only. – This configuration is loaded. The osinetbios command definition is not allowed. – The <value> Netbios NSAP is already defined as local NSAP. – The LSAP <value> is reserved or already defined. – The 0x<value> Subdomain is not defined in this configuration. – The 0x<value> Netbios Subdomain is not on a LAN adapter type. – Wrong parameter: <value>	– <The <value> Netbios NSAP has been defined. – The <value> Netbios LSAP has been defined. – The 0x<value> Netbios Subdomain has been defined.>

Unload Current Loaded Configuration

The Unload of the Current Loaded Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
Unload	– OSI Stack was not active, unload was not necessary. – Missing <pse> file. – Not able to unload <x–multiplexor>. Stop all OSI applications, and try again. – Unload failure, OSI driver kept loaded.	– Configuration name : <conf–name> – Stream modules loaded. – OSI driver unloaded. – OSI Stack unloaded.

Load of Last Selected Configuration

The Load of the Last Selected Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
Load	– A configuration must have been selected before. – OSI Stack already running, it must be unloaded previously. – Unload failure, OSI driver kept loaded. – Stream modules unloaded. – OSI driver unloaded. – OSI Stack unloaded. – File <name> is missing. – CAUTION: current configuration for MAX mbufs is : <size> Kb see 'System Environment' smit menu a recommended value for this configuration would be: <value> Kb. – CAUTION: current configuration for OSI Stack memory is : <size> Kb see 'Memory Management' smit menu a recommended value for this configuration would be: <value> Kb.	– Configuration name : <conf–name > (see also *Add a local NSAP*) (see also *Add a Subnet entry*) – Stream modules loaded. – OSI driver loaded. – OSI Stack loaded.

Change/Show Loaded Configuration

The Change/Show Loaded Configuration menu messages are shown in the following table:

Key Word	Errors	Responses
	General Errors: – Stack is not active.	
List Wait Entries	– Permission denied: lower layers not installed. – Error in receiving wait list. – Wait list is empty.	– < The list of the Wait Entries.>
List Dynamic RIB	– Permission denied: lower layers not installed. – Error in receiving dynamic RIB. – Dynamic RIB is empty.	– < The list of the Dynamic RIB entries.>
List Static RIB	– Permission denied: lower layers not installed. – Error in receiving static RIB. – Static RIB is empty.	– < The list of the Static RIB entries.>
Show a Dynamic RIB Entry	– Permission denied: lower layers not installed. – Error in RIB identifier: This RIB does not exist.	– < The selected RIB entry displayed in detail.>
Reset Dynamic RIB	– Permission denied: command for super user only. – Permission denied: lower layers not installed. – Dynamic RIB not reset.	– Dynamic RIB reset.
Reset Static RIB	– Permission denied: command for super user only. – Permission denied: lower layers not installed. – Static RIB not reset.	– Static RIB reset.
Display ES–IS status	– Permission denied: lower layers not installed.	– System type : <type> – Status : <state>
Show Memory buffer status	–	– current max memory = <value> Kb

Start Applications Support

The Configuration Management menu messages are shown in the following table:

Key Word	Errors	Responses
List of applications	OSI Stack is running, it must be unloaded previously.	# There is no application managed by SAS.
Change applications state	%s is not known by SAS.	%s already OFF %s already ON Start %s on osiload Do Not Start %s on osiload. \n\ usage: \n\ osisas -1 \n\ osisas -c Application [Application...] -s { ON | OFF } \n

Appendix D. OSI Applications & Communications Adapters Cross Reference

This appendix lists the OSI Applications supported on each type of Communications Adapter.

OSI Applications & Communications Adapters Cross Reference

The X25 HiSpeed WAN Comm. Adapter (single & 4–port) is supported by the following OSI Application:

- X25.3 API.

The Ethernet High–Performance LAN Adapter, Ethernet Standard Adapter and the X25 HiSpeed WAN Comm. Adapter (single & 4–port), are supported by the following OSI Applications:

- NetShare
- XTI API
- TPAD–HPAD
- UFT
- OTM 2A
- CPIC–SS.

The Ethernet High–Performance LAN Adapter, Ethernet Standard Adapter, X25 HiSpeed WAN Comm. Adapter (single & 4–port), Token Ring High–Performance Adapter and the FDDI Network Adapter are supported by the following OSI Applications:

- OSI API (ROSE & Session)
- XAP API
- ASN.1
- OSI–VT
- CPI–C OSI
- CPI–C PC side
- FTAM
- X.500 DS
- CX.400
- OSI–TP
- CMIP Agent for OSI Stack
- MMS

Applications User documentation is listed in the *AIX and Related Products Documentation Overview,* see Bibliography, on page B-1.

Glossary

This glossary contains the abbreviations, key–words and phrases that can be found in this document.

2LTP
Two–Level Transaction Processing.

ACB
Application Control Block.

ACSE
Association Control Service Element: Single consistent means for establishing and terminating all associations.

Adapter code
Address where the communications adapter is located inthe machine. The format is AA–BB–(CC).

ACSE
Association Control Service Element. Single consistent means for establishing and terminating all associations.

AFI
Authority Format Identifier.

AFNOR
Association Francaise de NORmalisation: French Standards Association.

AIO
Asynchronous I/O Extension.

API
Application Programming Interface: Functional interface allowing a high–level language application program to use specific data or functions of the operating system.

ARP
Abnormal Release Provider.

ASE
Application Service Element.

ASCII
American National Standard Code for Information Interchange.

ASE
Application Service Element.

AIX V3.2
International Business Machines Operating System: DPX/20 Operating System (Version 3.2) derived from AT&T UNIX System V.

BCD
Binary Coded Decimal.

BCS
Bull Cabling System.

Cache
High speed special buffer store.

CASE
Common Application Service Element.

CCB
Change Control Block.

CCITT
Consultative Committee on International Telegraphy & Telephone: United Nations Specialized Standards Group proposing recommendations for international telecommunications.

CD
Collision Detection.

CEN/CENELEC
Comité Européen de Normalisation ELECtronique: European Electronic Standards Commmittee.

CL
ConnectionLess.

CLI
Command Line Interface.

CLNP
ConnectionLess Network Protocol: Protocol where no acknowledgement is returned to the originating source.

CLNS
ConnectionLess Network Service.

Confirmation
Can be asked of a user before

execution of a command associated with a dialog (SMIT).

CO
Connection Oriented.

CONP
Connection Oriented Network Protocol.

CONS
Connection Oriented Network Service: Service with protocol provided by X.25.3 protocol.

COPP
Connection Oriented Presentation Protocol.

COSP
Connection Oriented Session Protocol.

COTP
Connection Oriented Transport Protocol.

COTS
Connection Oriented Transport Service.

CPI–C
Common Programming Interface for Communications: API allowing X/Open–compliant systems to communicate with systems implementing SNA Logical Unit type 6.2 (LU6.2) or XCP2 protocols.

CPR
Connect Presentation Reject.

CPU
Central Processing Unit.

CR
Connection Request.

CSDN
Circuit Switched Data Network.

CSMA–CD
Carrier Sense Multiple Access – Collision Detection.

CTP
Co–operative Transaction Processing.

CTX
Context.

CUD
Call User Data.

CUG
Closed User Group: Users who can inter–communicate within, but not outside of a closed group. Identification is inserted in the facilities field of call packets used to set up the Virtual Circuit.

DARPA
Defense Advanced Research Projects Agency: Department of Defense Agency (USA).

DAS
Dual–Attach Station (Network).

Datagram
The basic unit of information (a self–contained packet) that is passed across the Internet, containing the destination and source addresses.

Dataless workstation:
A machine with a local disk and local or remote file systems. The local disk may be used for a boot image, paging, or a local file system.

DCB
Data Control Block.

DCC
Data Country Code.

DCD
Data Carrier Detect.

DCE
Data Circuit–Terminating Equipment: Entry point to the network (X.25) including the modem and its interconnections.

DCM
Distributed Computing Model.

DCS
Defined Context Set.

Dialog
Window where parameter fields are entered before the execution of a required command (SMIT).

Dialog, ghost
Where there is no parameter and the command is executed immediately (SMIT).

DISC
SDLC frame used for DISConnection.

Diskless workstation
A machine with no disk and where boot images, paging space and all file systems reside on one or more servers.

DLL
Data Link Layer: Level between physical level and packet level, working according to the High–level Data Link Control procedure (HDLC).

DLSAP
Data Link Subnetwork Access Point.

DM
Disconnect Mode (SDLC frame).

DPF
Distributed Print Facility.

DPX/20
First generation Bull mainframe in UNIX environment.

DPX/20
Second generation Bull mainframe in UNIX environment.

DSA
Distributed Systems Architecture.

DSAC
Distributed Systems Administration & Control.

DSP
Domain Specific Part.

DTE
Data Terminal Equipment: Part of terminal installation which transmits and/or receives data.

DWM
Diskless Workstation Management: A set of utilities to help setup and manage a diskless workstation environment.

ECMA
European Computer Manufacturers' Association.

ES
End System: Final recipient system of the transport and upper OSI layer messages.

ES–IS protocol
End System – Intermediate System (Dynamic RIB).

ESH
End System Hello (PDUs).

Ethernet
A baseband LAN specification (IEEE 802.3) using the CSMA–CD technique.

FastPath
Simplified keystroke commands permitting SMIT functions to be quickly activated (IBM).

FCB
UFT Control Block.

FDDI
Communications adapter interface with a Fiber Distributed Data Network.

Field
Display dialog sub–window.

FIFO
First In First Out.

FRMR
FRame Reject (SDLC frame).

FTAM
File Transfer Access and Management: ISO file service function enabling user application processes to manage and access a file system.

FU
Functional Unit.

Functions
Communication products.

GAP
Subnet table.

Gateway
Software, linking two networks using different communication architectures. Gateway performs routing, conversion and relaying operations.

GCOS
General Comprehensible Operating System.

Git table
Memory Management Interface Table.

GOSIP
Government OSI Profile.

HCON
Host CONnection.

HDLC
High–level Data Link Control: Use of specialized series of bits to control data links in accordance with International Standards.

HPAD
Host PAD: Server side in the PAD client/server model.

ICB
Interface Control Block.

ICD
Initial Domain Identifier.

IDP
Initial Domain Part.

IEEE
Institute of Electrical & Electronic Engineers.

iFOR/LS
Information For Operation Retrieval / License System.

iniMAG
OSI Stack Administration initialization.

IP
Internet Protocol.

IPS
Internet Protocol Suite.

IS
Intermediate System: A relay system enabling data to be routed to the destination, or another intermediate system.

ISDN
Integrated Services Digital Network: Network supporting voice and non–voice communications.

ISH
Intermediate System Hello (PDU).

ISM
Integrated System Management.

ISO
International Standards Organization: Originator of Open Systems Interconnection reference model (ISO–IS 7498).

JFS
Journaled File System.

KDB
Kernel Debugger.

LAN
Local Area Network.

LAPB
Link–Access Procedure Balanced (also LAP): Link level elements used for data interchange (X.25 communications) between Data Circuit Terminating Equipment and Data Terminal Equipment operating in user classes of service 8 to 11, as specified in CCITT Recommendation X.1.

LFS
Logical File System.

LIFO
Last In First Out.

List
Mechanism which allows several possible values to be proposed in a parameter field. Number of values may vary. A list commands is executed to give possible values (SMIT).

LLC
Link Layer Control.

LLC
Logical Link Control: Protocol governing the assembly of transmission frames and their exchange between data stations, independent of the medium access control protocol.

LNSAP also **Local NSAPs**
Local Network Service Access Point: Access point to INTERNET Protocol (IP) services. Their names are used to identify the source and destination of messages.

LPP
Licensed Program Product.

LSAP
Link Service Access Point.

LVM
Logical Volume Manager: A flexible data storage system allowing the size of logical volumes

to span multiple physical volumes in a volume group; the data appearing in unbroken sequence to the user.

MAC
Medium Access Control.

MAD
Distributed Access Method: A set of primitives used to define relations between client and server processes which manage the communications services.

MAG
General Administrator Module.

Mandatory
Characteristic of a parameter field. If data is not entered in the field, the command of the dialog is not executed (SMIT).

MASK
A pattern of characters used as a control for other patterns of characters.

MCA
Micro Channel Architecture.

MHS
Message Handling Service.

MSDSG
Multi–system Distribution System Gateway.

MsgLifeTime
This parameter defines the validity period of the elementary Protocol Data Unit (PDU) transferred through the ISO 8473 protocol (CLNP). It determines the time–out before eliminating the packets lost because of non–efficient routing, or whose reference could be confused with more recent data. This period is measured by the number of passages in the Intermediate– and End–systems.

MTA
Message Transfer Agent.

MTS
Message Transfer System.

Multiselection
Characteristic of a parameter field for which several values can be entered (SMIT).

NCB
Network Control Block.

NCSC
National Computer Security Center (USA).

NET
Network Entity Title.

Network type
Type of outgoing subnetwork to be used (in accordance with subnet entries of the outgoing line to be used).

NFS
Network File System: Protocol developed by Sun Microsystems allowing users to directly access files on other systems in a network.

NIC
Network Information Center: Public distribution center for DARPA TCP/IP information.

NIS
Network Information System.

NIST
National Institute of Standards & Technology.

NMT
Network Management.

NPDU
Network Protocol Data Unit.

NSAP
Network Service Access Point: A chain of hexadecimal characters identifying the NSAP of a remote machine. It must be an even number of 40 characters maximum.

NSAP priority
Equivalent to the routing priority.

NSDU
Network Service Data Unit.

NSEL
Network SELector.

NUA
Network User Address: The 15–digit number that uniquely identifies an X.25 line.

ODM
Object Database Manager: A data

manager intended for the storage of system data (IBM).

OLTP
On–Line Transaction Processing.

OPP
Optional Program Product.

OSI
Open Systems Interconnection: Reference model defined in OS–IS 7498.

OTM
Open Terminal Manager.

Output
Window where the results of dialog commands are displayed. The standard output of commands are sent to this window (SMIT).

PAD
Packet Assembler Disassembler: Functional device enabling un–equipped Data Terminal Equipments to access a packet switching network.

PAVI
Videotex Access Point.

PC
Personal Computer.

PCI
Protocol Control Information.

PDU
Protocol Data Unit: Unit of protocol control information specified in the protocol of a given layer.

PHY
Physical Layer Protocol.

PICS
Protocol Implementation Conformance Statement.

PMD
Physical layer Medium Dependent.

POWER
Performance Optimization With Enhanced Risc.

Presentation
Presentation protocol: Set of actions and resources guaranteeing the presentation of the syntax of data during their transfer.

PRI
Primary Rate Interface.

PSDN
Packet Switched Data Network.

PSE
Portable Stream Environment.

PSTN
Public Switched Telephone Network.

PVC
Permanent Virtual Circuit: A Virtual Circuit (X.25) with a logical channel permanently assigned to it at each Data Terminal Equipment. A call establishing protocol is not required. Each PVC is identified with a name using the following syntax:
<site name><line number><PVC number> Where: Site name (4 characters) = identity of a unique X.25 subscription (valid if PVC number _ 1).
Line number (2 characters) = 01 (with 1 subscription per adapter).
PVC number (2 characters) = number between 1 and 8.

QOS
Quality Of Service.

QLLC
Qualified Logical Link Control: Data link control protocol enabling SNA to SNA communications over an X.25 network.

RAM
Random Access Memory: A storage device into which data can be written and subsequently read.

RAS
Reliability, Availability, Serviceability.

RCB
Request Blocks.

RD
ReDirect (PDU).

Relay – NSAP address
NSAP address of an intermediate system.

Remote SNPA type
Remote Sub Network Point of Attachment.

RFC
Request For Comments.

RIB
Routing Information Base: Network directory (library) which contains all the required routing information to remote NSAPs. For each (group of) NSAPs, it gives the remote SNPA and local subnetwork to use. The SNPA is found by using the Subnet table.

RIB, Static, “variable”
STATICV: the entry is part of the “variable” RIB on disk loaded into CPU memory only on express request of the internetwork protocol (CLNP). The allocated CPU memory is called the “variable” Static RIB.

RIB, Static, “fixed”
STATICF: the entry is part of the “fixed” RIB on disk automatically loaded into CPU memory at the initialization of the OSI Stack. The allocated CPU memory is called the “fixed” Static RIB.

Ring
Loop function which allows several predefined values to be proposed in a parameter field (SMIT).

ROSE
Remote Operations Service Element.

RTS
Reliable Transfer System.

SAP
Service Access Point.

SAS
Stack Applications Support.

SAS
Single–Attach Station.

SCB
Session Control Block.

SCO
Santa Cruz Operations.

SCX
DPX/20 Communications System.

SDLC
Synchronous Data Link Control: Control using commands to regulate the transfer of data over a communications line.

SDU
Service Data Unit.

Selector
Window where a function parameter is entered with the possibility to: select a subsequent dialog or find default values for parameters of subsequent dialogs (SMIT).

Session
Session protocol: Virtual relationship permitting communications between two network addressable units.

SID
Specifications ISO/DSA.

SMIT
System Management Interface Tool (IBM): Menu–driven, resident command–building system management facility.

SMT
Station Management.

SMTP
Shared Memory Transport Protocol.

SNA
Systems Network Architecture.

SNAP
Service Network Access Point.

SNDCF
SubNetwork Dependent Convergent Function.

SNPA
Sub Network Point of Attachment: Information for accessing the system within the domain (Transpac or Ethernet address).

SPECFS
Special File System.

SPI
Subsequent Protocol Identifier: Used for routing incoming calls.

SPOT
Shared Product Object Tree: The */export/exec* directory on a server which contains a client’s */usr* file system.

SSAP
Session Service Access Point.

SSDU
Session Data Unit.

SSEL
Session SELector.

STP
Shielded Twisted Pair.

Stanza file
Description file of screen objects used in Object Data Manager to update the SMIT database (SMIT).

Subdomain
Logical Name associated to a physical network, linking RIB information and the Subnet table. Linking the local system and the next system to be reached. Decimal or hexadecimal character chain (value given using a 32 bit format).

Subnet entry identifier
Code with CN format where: C is a character representing the Network Type: E for Ethernet: R for Token Ring: X for X.25. N is number in the range 0 to 9, identifying a communications adapter from another of the same type.

Subnet table
Describes all the available network access on a station. Networks are identified by subdomain names, each associated with a physical address (SNPA).

Super–user
A System Administrator with unrestricted authority to access and modify any part of the Operating System.

SVC
Switched Virtual Circuit: Requested by a virtual call and released when the call is cleared.

SYSLFS
VFS switch layer.

TA
Terminal Adapter.

TCB
Transport Control Block.

TCP
Transport Control Protocol: Protocol used in ARPA Internet (U.S. Department of Defense standards for inter–networks).

TCP–IP
TCP and IP are the two fundamental protocols of the Internet protocol suite. (Acronym for this suite). TCP provides reliable transfer of data, while IP transmits.

TCSEC
Trusted Computer System Evaluation Criteria.

Token Ring
Access procedure used with a sequential topology.

TP
Transactional Processing

TPAD
Terminal PAD: Client side in the PAD client/server model.

TPDU
Transport Protocol Data Unit.

TPI
Transport Provider Interface.

TPISES
Transport Provider Interface for SESsion layer: OSI driver module which allows the Stack session layer to communicate with the TP1006 driver.

TRANSPAC
French public packet–switched network offering connections in packet mode (X.25) or character mode (X28–X3).

TSAP
Transport Service Access Point.

UA
User Agent.

UCB
User Control Block.

UFT
Unified File Transfer.

UNIX
Portable operating system, implemented in “C” language.

US GOSIP V1
U.S. Department of Commerce – Federal Information Processing Standards 146. August 1988 – Government Open Systems Interconnection Profile – Version 1.

UTP
Unshielded Twisted Pair.

VC
Virtual Circuit: A logical end–to–end X.25 transmission channel.

VFS
Virtual File Service: A remote file system mounted so that it is accessible to the local user.

VT
Virtual Terminal.

WAN
Wide Area Network: Network providing communications capability in geographic areas larger than served by Local Area Networks.

Window
An area of a display screen with visible borders within which information is displayed.

XCB
General term for different control blocks corresponding to the OSI levels, SCB, TCB, NCB, ...

XCP2
eXtended Cooperative Protocol level 2.

XTI
X/Open Transport Interface Definition.

Xwindows display interface
A software graphical user interface environment based on AIX.

X.21
CCITT recommendation defining a synchronous interface for public data networks.

X.25
CCITT recommendation defining an interface for connection to a packet–switched network with virtual circuit service. Defines the three lowest ISO layers: Electrical interface, HDLC procedure application, Packet structure.

X.25.3
Packet level protocol in layer three of X.25.

HiSpeed WAN Comm.
HiSpeed WAN Comm.. Adapter (x4 = 4–port, x1 = single port).

X.121
The 121st CCITT recommendation in the X series, defining a naming convention for the network user address.

X.400
Message handling service. CCITT recommendation defining the message handling services which can be provided by telecommunications authorities to their subscribers enabling them to exchange messages in the store and forward mode.

X.500
Open System directory management service applying CCITT recommendations.

Index

D

M

N

O

P

Q

R

S

T

U

W

X

Vos remarques sur ce document / Technical publication remark form

Titre / Title : Bull DPX/20 OSI Services Reference Manual

Nº Reférence / Reference Nº : 86 A2 05AQ 02

Daté / Dated : June 1996

ERREURS DETECTEES / ERRORS IN PUBLICATION

AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement
Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.
If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : ________________________ Date : __________

SOCIETE / COMPANY : ________________________

ADRESSE / ADDRESS : ________________________

____

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

Bull Electronics Angers S.A.
CEDOC
Atelier de Reprographie
331 Avenue Patton
49004 ANGERS CEDEX 01
FRANCE

PLACE BAR CODE IN LOWER
LEFT CORNER

Bull Electronics Angers S.A.
CEDOC
Atelier de Reprographie
331 Avenue Patton
49004 ANGERS CEDEX 01
FRANCE

ORDER REFERENCE
86 A2 05AQ 02

Utiliser les marques de découpe pour obtenir les étiquettes.
Use the cut marks to get the labels.

DPX/20
AIX
OSI Services
Reference Manual

86 A2 05AQ 02

DPX/20
AIX
OSI Services
Reference Manual

86 A2 05AQ 02

DPX/20
AIX
OSI Services
Reference Manual

86 A2 05AQ 02