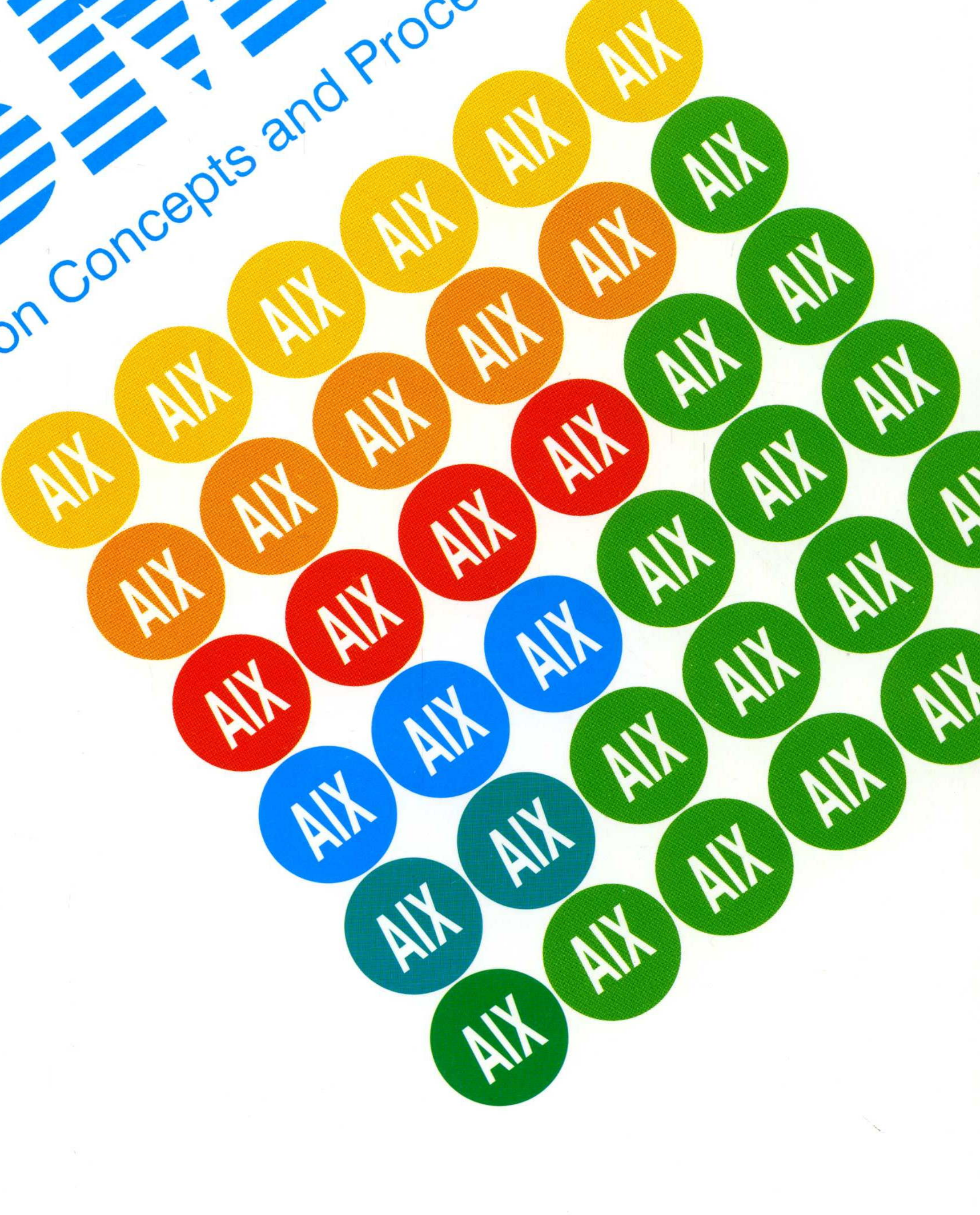
AIX Version 3 for
RISC System/6000™

Communication Concepts and Procedures
Volume 2



SC23-2203-00

First Edition (March 1990)

This edition of the *AIX Communication Concepts and Procedures for IBM RISC System/6000* applies to *Version Number 3* of the IBM AIX Base Operating System licensed program, AIX SNA Aervices/6000, AIX 3278/79 Emulation/6000, and AIX Network Management/6000 and to all subsequent releases of these products until otherwise indicated in new releases or technical newsletters.

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS MANUAL "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

IBM does not warrant that the contents of this publication or the accompanying source code examples, whether individually or as one or more groups, will meet your requirements or that the publication or the accompanying source code examples are error-free.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time.

It is possible that this publication may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country. Any reference to an IBM licensed program in this publication is not intended to state or imply that you can use only IBM's licensed program. You can use any functionally equivalent program instead.

Requests for copies of this publication and for technical information about IBM products should be made to your IBM Authorized Dealer or your IBM Marketing Representative.

A reader's comment form is provided at the back of this publication. If the form has been removed, address comments to IBM Corporation, Department 997, 11400 Burnet Road, Austin, Texas 78758–3493. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

The Network File System (NFS) was developed by Sun Microsystems, Inc.

The Rand MH Message Handling system developed by the Rand Corporation and the University of California.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. We acknowledge the following institutions for their role in its development: the Electrical Engineering and Computer Sciences Department at the Berkeley Campus.

Trademarks and Acknowledgements

The following trademarks and acknowledgements apply to this book:

AIX is a trademark of International Business Machines Corporation.

AIXwindows is a trademark of International Business Machines Corporation.

Apollo is a trademark of Apollo Computer, Inc.

Hayes is a registered trademark of Hayes Microcomputer Products, Inc.

IBM is a registered trademark of International Business Machines Corporation.

NCK is a trademark of Apollo Computer, Inc.

NCS is a trademark of Apollo Computer, Inc.

Network Computing Kernel is a trademark of Apollo Computer, Inc.

Network Computing System is a trademark of Apollo Computer, Inc.

Network File System and NFS are trademarks of Sun Microsystems, Inc.

RISC System/6000 is a trademark of International Business Machines Corporation.

SNA 3270 is a trademark of International Business Machines Corporation.

UNIX was developed and licensed by AT&T and is a registered trademark of AT&T Corporation.

About This Book

This book contains conceptual and procedural information about a variety of communications facilities and applications.

Who Should Use This Book

This book is intended for people who want to use communications applications and services and who want to perform system management tasks that involve communication within a network.

How to Use This Book

The chapters of this book are divided into concepts for people who want to use the system or network, concepts for people who manage a system or network, and procedures for various tasks. The concept chapters are divided into overviews and subsequent information. The procedures are arranged in logical order and contain both using and system administration tasks.

Overview of Contents

The Communication Concepts and Procedures book contains information about several types of communication facilities. Part one contains information about emulators available for the RISC System/6000, including Asynchronous Terminal Emulation (ATE), 3278/79 Emulation (EM78), and the AIX 3270 Host Connection/6000 (HCON) program.

- Asynchronous Terminal Emulation (ATE) allows your system to emulate an asynchronous terminal attached to a remote computer. This chapter contains information about setting up ATE on your system and customizing it to meet your needs.
- The AIX 3278/79 Emulation/6000 Licensed Program (EM78) allows your system to emulate a terminal attached to an IBM System/370 host computer. This chapter contains information to help you install, configure, and customize EM78 on your system.
- The AIX 3270 Host Connection Program/6000 Licensed Program (HCON) allows your system to emulate a terminal or printer attached to an IBM System/370 host computer. This chapter contains information on installing HCON, setting up HCON users and profiles, and customizing HCON on your system. Also included is information to help you maintain HCON after it is in use.

Part two contains information about programs and facilities to use for network communications, including applications for remote terminal use (BNU), Mail and Message Handler facilities, applications for sharing files across a network (NFS/NIS and NCS), and applications for monitoring a network (Alerts and SNMP **xgmon**).

- The Basic Networking Utilities programs (BNU) are the AIX version of the UNIX-to-UNIX Copy Program (UUCP). They allow you to transfer files to and execute commands on remote systems, and to let remote systems transfer files to and execute command requests on your system. They also provide remote mail facilities. This chapter contains information to help you set up BNU and monitor its activities. It includes information on defining a BNU/UUCP network, scheduling access and contact times, and securing your system against unauthorized access by other systems that use BNU or UUCP.
- The Generic Data Link Control (GDLC) is a generic interface definition that allows both application and kernel users to have a common set of commands to control DLC device managers within the AIX Version 3 system. This chapter contains information on how to add, change, list, and remove data link controls.

- The Mail system is a general purpose, internetwork mail-routing facility. This system is not tied to any one transport protocol. It relays messages from one user to another across system and network domain boundaries. While processing the messages, the mail system can do a limited amount of message-header editing to put the message into a format that is appropriate for the receiving domain. This chapter provides information for both the daily use and management of Mail.

- The Message Handler (MH) package enables you to create, distribute, receive, view, process, and store messages. This chapter contains information to help you manage, customize, and use the MH package. Note that the MH package does not provide a message transport facility; instead, it relies on the transport facilities associated with the **sendmail** command.

- The Network Computing System (NCS) allows you to distribute processing tasks across resources in a network or internet by maintaining databases that control the information about the resources. NCS consists of three components: the Remote Procedure Call runtime library, the Location Broker, and the Network Interface Definition Language compiler. This chapter provides a brief introduction to the working of NCS and its components as well as detailed information on how to configure NCS.

- The Network File System (NFS) is a distributed file system that allows you to access files and directories remotely on a network as if they were local. The IBM NFS network information service (NIS) is a network service used to distribute system information on networked hosts. This chapter contains information to help you install, configure, and maintain NFS and NIS on your system.

- The AIX Network Management/6000 Licensed Program (Alerts program and the **xgmon** command) is a network management program for monitoring TCP/IP networks. It assists you in monitoring the status of all the machines on a network and helps you anticipate network problems. This chapter contains information about the Alerts monitoring application and how to start, use, maintain, and customize the **xgmon** program.

Part three contains information about network connectivity and includes Systems Network Architecture/6000, Transmission Control Protocol/Internet Protocol (TCP/IP), and X.25 Communications.

- TCP/IP is a communications subsystem that allows you to set up local area and wide area networks. TCP/IP allows you to transfer files between systems, log in to remote systems, run commands on and print files to remote systems, and communicate interactively or through mail with remote users. This chapter contains information to help you configure and customize TCP/IP. Also included is information to help you manage a network using TCP/IP. TCP/IP provides basic network management capability.

- Systems Network Architecture (SNA) is a specification that formally defines the functional responsibilities for components of a data communications system and specifies how those components must interact. This chapter contains information to help you to set up and customize SNA for your system. It includes information on SNA concepts, AIX SNA services, installing AIX SNA services, configuring a network using SNA, controlling SNA on the local system, and using the SNA extensions to the SMIT Interface.

- The X.25 communications service provides intercommunication between systems. It is particularly useful for communicating with people using different computer systems and for applications that access public data bases. This chapter includes information about planning, installing, configuring and customizing your X.25 network.

Highlighting

The following highlighting conventions are used in this book:

Bold	Identifies commands, keywords, files, directories, and other items whose names are predefined by the system.
Italics	Identifies parameters whose actual names or values are to be supplied by the user.
`Monospace`	Identifies examples of specific data values, examples of text similar to what you might see displayed, examples of portions of program code similar to what you might write as a programmer, messages from the system, or information you should actually type.

Related Publications

The following books contain information about or related to communications:

- *AIX Communications Programming Concepts for IBM RISC System/6000*, Order Number SC23–2206.
- *AIX General Concepts and Procedures for IBM RISC System/6000*, Order Number SC23–2202.
- *AIX Commands Reference for IBM RISC System/6000*, Order Number SC23–2199.
- *AIX Files Reference for IBM RISC System/6000*, Order Number SC23–2200.
- *IBM RISC System/6000 General Information and Planning Information Kit*, Order Number GK2T–0237.

Ordering Additional Copies of This Book

To order additional copies of this book, use Order Number SC23–2203.

Table of Contents

Part 1. Emulators

Part 2. Network Facilities

Part 3. Network Connectivity

Part 3. Network Connectivity

Chapter 13: AIX SNA Services/6000

Chapter 13 contains all the information necessary to understand, install, configure, and maintain AIX SNA Services/6000. Included is a comprehensive introduction to AIX SNA Services/6000, a thorough and complete section on how to configure AIX SNA Services/6000 to work with your network and application programs, and a procedural section designed to aid in starting and stopping AIX SNA Services/6000 and SNA Services/6000 attachments and connections. Information about network security and about accessing network information can be found as well. Profile forms have been included to facilitate planning for the configuration of your system. These forms complement the aforementioned configuration section and should be used as a tool in the configuration process. Finally, a short section containing information and sample profiles for LU 6.2 and LUs 1, 2, and 3 connections is included.

Introducing IBM AIX Systems Network Architecture Services/6000

Introducing AIX SNA Services/6000 introduces you to the concepts and terminology of SNA and shows how AIX SNA Services/6000 follows the general model for SNA systems. It also provides more detailed description of the SNA features that AIX SNA Services/6000 supports. Introducing AIX SNA Services/6000, together with the SNA documents referenced in the preface to this book, provides necessary background to help you configure and operate AIX SNA Services/6000.

Introduction

Systems Network Architecture (SNA) is a specification that formally defines the functional responsibilities for components of a data communications system and specifies how those components must interact. In an SNA structure, all *nodes* (linked elements) follow these definitions.

The AIX Systems Network Architecture (SNA) Services implements SNA as a part of IBM's AIX Operating System. It provides a unified communications system that is consistent across a wide range of data processing systems. This service also relieves you of concern for many network resource control and management requirements. In most cases, you can work with an application that uses AIX SNA Services/6000 without being aware that AIX SNA Services/6000 is working for you.

AIX SNA Services/6000 is a set of programs and data files (see the table below) woven into the fabric of the operating system to provide nearly transparent access to the resources of an SNA network. Programs access the network through a device driver, **/dev/sna**, so that access to the network operates the same as access to any other AIX Operating System I/O (input/output) device. AIX Operating System treats each attachment or connection to a remote system as if it were a file in a directory called /**dev/sna**. This method allows AIX SNA Services/6000 to support the *sequential I/O model* of the operating system for all accesses to network resources. Support of this model simplifies access to the network, allows programs to be designed for portability, and even allows programs that use standard input and standard output to use network resources using redirection.

This unified approach to accessing the network helps provide the following advantages:

- A programmer can write a program that uses the network without knowing:
  - Details of SNA
  - Details of any data communications protocol
  - How to configure communications resources in a network.
- A program that operates with one supported network environment can operate with all supported network environments.
- A program that operates in this network environment is shielded from either changes that occur due to updates in the operating system or changes in a network protocol.

SNA Directories and Files

The following table lists and defines important SNA files and directories:

SNA Directories and Files

Path Name	Function
/usr/bin/	
adcs	Command that starts the ADCS emulator program.
chsnalias	Command that changes the description of an alias in the SNA configuration database.
chsnaobj	Command that changes the description of a currently defined profile in the SNA configuration database.
chsnapw	Command that changes the password for the SNA configuration database.
exportsna	Command that exports SNA configuration database profiles to a file or printer in stanza format.
gensnakey	Command that generates security keys.
hcp	Command that starts the HCP emulator program.
hcps	Command that starts the HCP SUP command program.
importsna	Command that adds profiles to the SNA configuration database, using a file created earlier by the **exportsna** command as its standard input.
linktest	Command that tests the data link with a remote system.
lssnaobj	Command that lists all SNA configuration database profiles that meet a user-specified criteria.
lu0	Command that starts the LU0 server program.
lu0config	Command that starts the LU0 configuration program.
lu0pass	Command that starts the LU0 passthrough program.
lu0sndmsg	Command that sends commands to the LU0 server.
mksnalias	Command that adds an alias to the SNA configuration database for a specified profile.
mksnaobj	Command that adds a profile to the SNA database.
mksnapw	Command that adds a password to the SNA configuration database.

Path Name	Function
qrysnaobj	Command that returns the value of a field specified by the user for a given profile of a given profile type.
rmsnalias	Command that removes an alias from the SNA configuration database for a specified profile.
rmsnaobj	Command that deletes one or more aliases or profiles and all of the object's aliases from the SNA configuration database.
rmsnapw	Command that deletes the SNA password.
stophcp	Command that stops the HCP emulator program.
stophcps	Command that stops the HCP SUP command program.
verifysna	Command that verifies cross-dependencies within the SNA configuration database.
/usr/lpp/sna/bin/	
luxcnos	Service application source program for changing the number of sessions.
luxcnost	Service application target program for changing the number of sessions.
luxcps	Contains both connection point and physical unit services, as well as the control to start and stop links and attachments.
luxcr	Command router is the parent starter process of SNA and provides SNA resource management function.
luxihd	The SNA control daemon.
luxlns	The LU network services component (LNS) initiates and terminates LU-LU sessions in response to requests from the resource manager or the remote LU. LNS also activates and deactivates CP-LU sessions
luxlrm	The LU resource manager.
luxrqcp	Service application source program that exchanges CP capabilities.
luxcpcap	Service application target program that exchanges CP capabilities.
peu	Command that creates a default set of profiles in the **/usr/lpp/sna/objrepos** directory.
sna_update.awk	Command that updates AIX SNA Services/6000 version 2.2.1 profiles (in stanza format) to version 3.1 profiles (in stanza format).
luxicfg	Program used by SNA install procedure.
luxirest	Program used by SNA install procedure.
luxlogdm	SNA Services internal error logging daemon.
/usr/lpp/sna/objrepos/	Directory containing AIX SNA Services/6000 configuration database.
alias	File containing SNA profile information for the alias object class.
attachment	File containing SNA profile information for the attachment object class.

Path Name	Function
connection	File containing SNA profile information for the connection object class.
control_pt	File containing SNA profile information for the control point object class.
local_lu	File containing SNA profile information for the local LU object class.
log_*Link*	File containing SNA profile information for the logical link object class.
lu_reg	File containing SNA profile information for the generic LU address registration object class.
mode	File containing SNA profile information for the mode object class.
mode_list	File containing SNA profile information for the mode list object class.
phy_*Link*	File containing SNA profile information for the physical link object class.
sna	File containing SNA profile information for the SNA object class.
tpn_list	File containing SNA profile information for the transaction program name list object class.
transact	File containing SNA profile information for the transaction program name object class.
/usr/lpp/sna/samples/	Directory containing the **sendto.c** and **rcvfrom.c** programs, sample profiles that can be modified for use with the **sendto.c** and **rcvfrom.c** programs, and sample profiles that can be used to connect to various other systems and LU types.
LU123.prof	Sample LU 1, 2, 3 host connection profiles.
cics.prof	Sample cics host connection profiles.
rcvtrn.prof	Sample file transfer profiles.
s36.prof	Sample S/36 connection profiles.
s38.prof	Sample S/38 connection profiles.
sendto.c	C language source code for sample file transfer send program.
rcvfrom.c	C language source code for sample file transfer receive program.
/usr/lpp/lu0/	
lu0api.c	File containing C source for the **lu0api** subroutine.
adcsapi.h	Header file containing ADCS user application API definitions.
lu0api.h	Header file containing LU0 API definitions.
lu0apis.h	Header file containing the **cmd4680** command file record formats.
lu0conf.h	Header file containing configuration file definitions.
lu0.cnf	Default LU0 configuration file.
/usr/lpp/lu0/bin/	
adcsemul	Binary object file for ADCS emulator program.
hcpemul	Binary object file for HCP emulator program.

Path Name	Function
hcpsuser	Binary object file for HCP command.
killhcp	Binary object file for HCP command.
killhcps	Binary object file for HCP command.
lu0confg	Binary object file for LU 0 configurator.
lu0pthru	LU 0 passthru program.
lu0server	LU 0 interface processor.
lu0smsg	Binary object file for issuing background **lu0** server commands.
/usr/lpp/msg/En_US/	
sna_EN.cat	SNA message catalog.
/usr/include/	
luxsna.h	Header file containing constant and structure definitions for AIX SNA Services/6000 subroutines.
lu0.h	Header file containing LU0 common definitions.
/usr/lib/	
libsna.a	Library of subroutines that provide access to AIX SNA Services/6000 resources from a C language program.
liblu0.a	Library of subroutines that provide access to AIX SNA Services/6000 LU0 support from a C language program.
/etc/	
rc.sna	Shell script used for automatic start up of SNA Services.
/etc/methods/	
defsna	File containing the define SNA method used to define the SNA kernel extensions
cfgsna	File containing the configure SNA method used to configure the SNA kernel extensions.
udefsna	File containing the undefine SNA method used to undefine the SNA kernel extensions
ucfgsna	File containing the unconfigure SNA method used to unconfigure the SNA kernel extensions.
lscfgsna	File containing the SNA list configuration method.
/etc/drivers/	
sna_sysx	This file contains the binary object code for the SNA Services kernel functions (with the exception of those in the **sna_pin** file).
sna_pin	This file contains the binary object code for the portion of the SNA Services kernel functions that must reside in pinned memory.

Figure 3. SNA Services Components

AIX SNA Services/6000 Structure

The following paragraphs describe the major SNA Services/6000 components and the part they play in the operation of the SNA network. The figure on page 13–9 illustrates the structure of AIX SNA Services/6000 and how it's components relate to other parts of the AIX Operating System.

Application Program

The application program, sometimes called the *transaction* program, is *not* supplied with AIX SNA Services/6000 but uses one of the programming interfaces to AIX SNA Services/6000 to access a network. The program may be a product sold separately by IBM, one purchased from another software supplier, or a program that you write to accomplish a special task.

The application program does not need to account for network protocol or other network considerations. It simply uses the supplied C language library routines or AIX Operating System subroutines to make requests to AIX SNA Services/6000, and then waits for a response. AIX SNA Services/6000 handles all the details required to communicate with a configured network node. Refer to Writing Transaction Programs in *Communications Programming Concepts* for simplified examples of C language programs that use AIX SNA Services/6000.

System Resource Controller

The System Resource Controller provides support through commands and subroutines for starting, stopping and querying the status of AIX subsystems. Refer to the System Resource Controller Overview in *General Concepts and Procedures* for more information on these commands and subroutines

AIX SNA Services/6000 Library Subroutines

The SNA library, **libsna.a**, contains a set of subroutines that allow an application program to:

- Establish a network connection with one or more remote application programs, exchange data with those programs, and disconnect from them. These are called AIX SNA Services/6000 library subroutines.
- Send and receive Network Management Vector Transport (NMVT) data.

AIX SNA Services/6000 Subroutines

Using these subroutines, the application program can interact directly with the network without having to account for the details of network protocols. The subroutines provide a standard format for communicating with the network that helps ensure portability of the program to other systems and independence of the program from the type of network protocol used. The network type of the program must be supported by AIX SNA Services/6000. The library services translate the subroutines into the required AIX Operating System subroutines to perform the following requested functions:

- Initializing a connection to a remote node
- Creating a conversation between a local program and a remote program
- Transmitting data between the local and the remote program
- Monitoring and controlling the conversation
- Ending the conversation and the connection.

Refer to AIX SNA Services/6000 Library Subroutines in *Communications Programming Concepts* for descriptions of the SNA library subroutines.

Network Management Vector Transport (NMVT) Subroutines

The following functions are provided by this set of subroutines:

- Getting the status of an SSCP-PU session
- Sending NMVT data on an SSCP-PU session
- Receiving NMVT data on an SSCP-PU session.

Refer to Subroutines for Network Management (System Services Control Point Subroutines) in *Communications Programming Concepts* for descriptions of the network management subroutines.

Standard I/O Library Subroutines

The standard I/O library, a part of the **libc.a** library, contains a set of subroutines that allow an application program to write data to and read data from the devices assigned as standard output and standard input. By assigning an SNA connection to either standard input or standard output, programs that use the standard I/O library can input or output data to that connection.

For example, the AIX Operating System **cat** command uses standard input and standard output. If AIX SNA Services/6000 is running and suitable connection profiles have been defined on both the remote and local systems, the following steps can transfer the **testfile** file from the local system to the remote system. Both connections must use the limited interface. For this example, the connection profiles are called remote on the local system and local on the remote system.

1. Define a transaction program on the remote system (see the following example) to receive input from the local system. You must create both a transaction program name (TPN) profile and a TPN list profile on the remote system to define this program.
2. Define the remote transaction program to the local system by creating an RTPN profile and an RTPN list profile that names the program that you added to the remote system.
3. Enter the following command on the local system:

   ```
   cat <testfile >/dev/sna/remote
   ```

   This command reads the contents of the **testfile** file and sends the file on the network to the remote system.

As soon as the command is entered, the local system opens the connection to the remote system and sends the **testfile** file to the remote system. The remote program is activated by AIX SNA Services/6000. The program receives the file and places the contents in the **newfile** file (as defined by the remote program). The following is a sample program listening for a remote system:

```
main( argc, argv )
int argc;
char **argv;
{
      char PATH[60];
      strcpy( PATH, "cat < /dev/sna/" );
      strcat( PATH, argv[2] );
      strcat( PATH, "> newfile" );
      system( PATH );
}
```

Figure 4. Sample Listening Program for Remote System

AIX Operating System Subroutine Calls

AIX Operating System subroutine calls are the basis for all communication between an application program and the operating system. The previously described SNA Services/6000 library subroutines are ultimately converted to equivalent AIX Operating System subroutine calls before they perform their functions.

AIX Operating System subroutine calls access the network architecture through the SNA device driver (/**dev/sna**). This device driver is a multiplex device driver that allows an extended path name to specify which of the multiple devices is being addressed by the subroutine. The extended path name takes the form:

```
/dev/sna/Identifier
```

The device driver interprets the identifier to mean one of the defined connections. In that way, the device driver behaves like a directory that contains all SNA connection device drivers. For example, if you have defined a connection profile that specifies the connection to a remote system, the following subroutine opens that connection for reading and writing (O_RDWR):

```
open(/dev/sna/remote, O_RDWR);
```

AIX SNA Services/6000 provides two kinds of subroutine support, the limited interface and the extended interface. Both interfaces use standard AIX Operating System subroutines.

SNA System Resource Manager

The resource manager function for AIX SNA Services/6000 receives requests from the System Resource Controller and interacts with the other LU components to coordinate the use of sessions and conversations for the LU. Refer to the System Resource Controller Overview in *General Concepts and Procedures* for more information on the SRC. Some of the tasks that the resource manager performs include:

- Coordinating the starting of connections and attachments
- Coordinating status reporting
- Managing the activation or deactivation of a session
- Managing the security of a conversation.

The resource manager is closely tied to both LU (logical unit) services and PU (physical unit) services.

LU Services

LU services responds to commands issued through the System Resource Controller to control SNA resources by providing status information or by sending the required network commands to establish or end a session with a remote LU. LU services also provides the control required to support the local half session. Refer to the System Resource Controller Overview in *General Concepts and Procedures* for more information on the SRC.

PU Services

PU services provides control of the physical configuration and the resources of the local system. It is responsible for the status of the data link and for maintaining an attachment with a remote system.

The figure on page 13–9 illustrates the structure of AIX SNA Services/6000 and how it's components relate to other parts of the AIX Operating System.

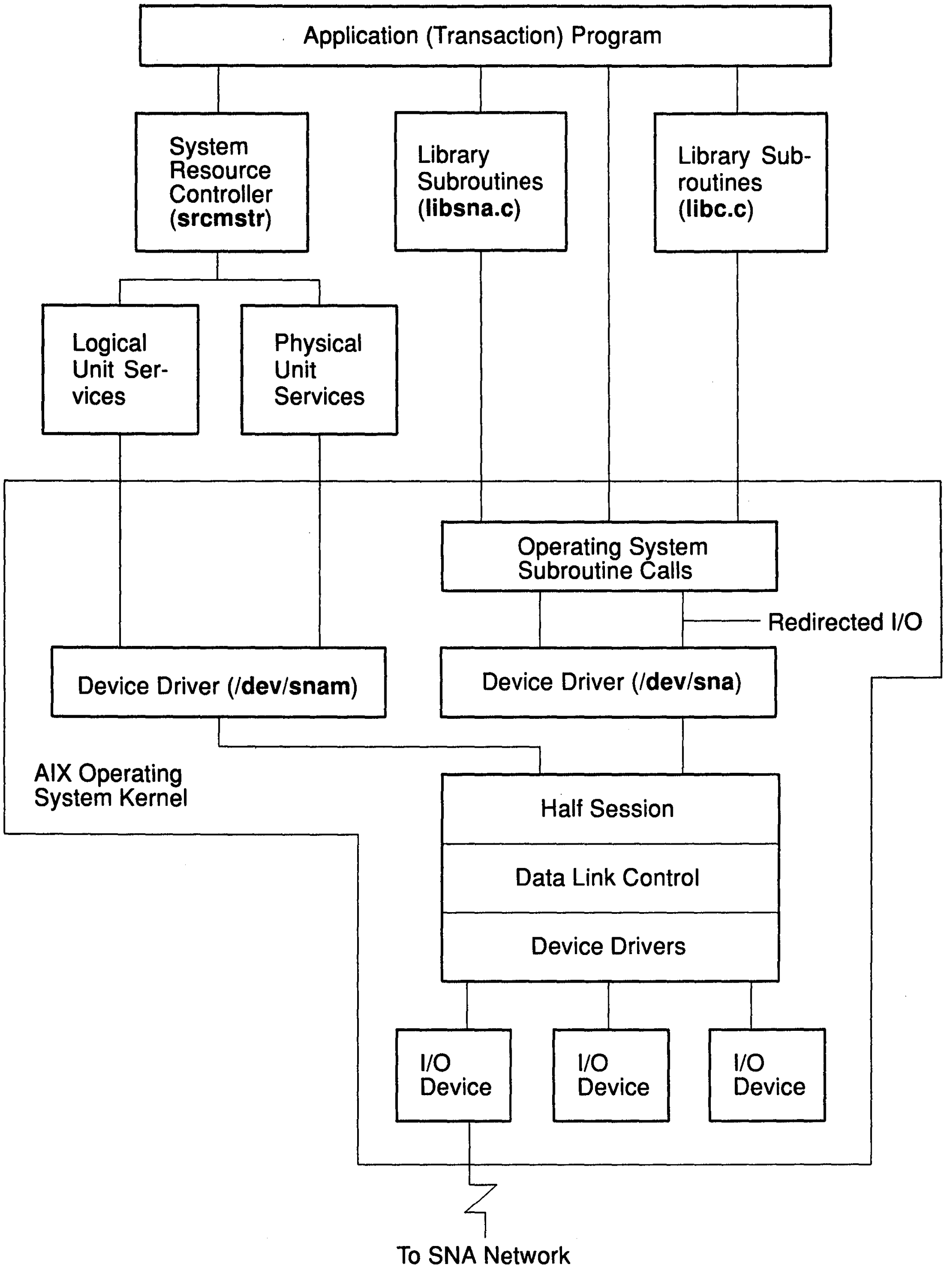

Application (Transaction) Program
System Resource Controller (srcmstr)
Library Subroutines (libsna.c)
Library Subroutines (libc.c)
Logical Unit Services
Physical Unit Services
Operating System Subroutine Calls
Redirected I/O
Device Driver (/dev/snam)
Device Driver (/dev/sna)
AIX Operating System Kernel
Half Session
Data Link Control
Device Drivers
I/O Device
I/O Device
I/O Device
To SNA Network

Figure 5. AIX SNA Services/6000 System Components

SNA Components

AIX SNA Services/6000 implements two SNA components that control the operation of the local node in the network. These components are:

- Physical Unit (PU)
- Logical Unit (LU).

Each component has a special assigned function in the network. Because each of these components can be separately addressed by other members of the network, these components are called *network addressable units* (NAUs). Combinations of these NAUs make up a single *node* of an SNA network. A node iszx Each SNA node has both a PU and an LU component.

Physical Unit (PU)

This component controls the physical resources of the node. Physical resources include the data links that connect the node to the network, storage, and input/output devices. All nodes in the network must have a PU. Depending upon the function provided by the PU, it is classified as one of the following PU types (PUT):

PUT 2.1 A converged peripheral node that has limited addressing and path-control routing capabilities. This node type provides general connectivity to other SNA nodes and supports parallel sessions, multiple sessions for each LU, primary and secondary LUs, and multiple links for each node.

PUT 4 or **PUT 5** A subarea node that provides network-wide addressing and control data flow within a subarea (the subarea node and all peripheral nodes connected to it). PUT 4 does not contain an SSCP component; PUT 5 does. AIX SNA Services/6000 *cannot* perform the functions of a PUT 4 or a PUT 5 subarea node.

Logical Unit (LU)

This component provides the interface to the network for the end user. It provides protocols that allow end users to communicate with each other and with other components in the network. Depending upon the protocol that the LU implements, it is classified as one of the following LU types:

LU 0, **LU 1**, **LU 2**, and **LU 3**

These LU types communicate with a host computer in a primary-to-secondary relationship. The host computer, as the primary node, controls the data interchange between the two nodes. AIX, as the secondary node, responds to the primary node and provides input or output services. These LU types differ mainly in the kind of input and output services they provide:

LU 0 The AIX implementation of this protocol provides and manages communication between devices associated with the retail industry (for example, point-of-sale terminals). It provides primary and secondary LU support through an Application Programming Interface (API).

LU 1 This protocol manages many input/output devices associated with the LU, such as printers, card readers and punches, storage devices, and an operator console.

LU 2 This protocol emulates an IBM 3270 data terminal connected to the LU, allowing keyboard input, display output, and file transfer, using the SNA 3270 data stream.

LU 3 This protocol emulates an IBM 3270 attached printer connected to the LU, allowing for printed output, using the SNA 3270 data stream.

LU 6.2 This protocol provides Advanced Program-to-Program Communications (APPC) for communications between two programs on a peer-to-peer basis instead of a primary-to-secondary one. This protocol allows programs on the AIX node (such as a peripheral node, PUT 2.1) to communicate with programs running on a subarea node (PUT 4 or PUT 5) or on another peripheral node (PUT 2.1).

When communicating with an LU 6.2 host computer, LU 6.2 can operate as a dependent node.

Describing the Data Exchange Environment

The process of transferring data from an application program at one node to an application program at another node involves joining parts of the network at different levels. The process begins by establishing physical hardware paths between two nodes and ends with the two programs exchanging information. These levels are shown in the following figure. An explanation of a connection, an attachment, a session, and a conversation follow.

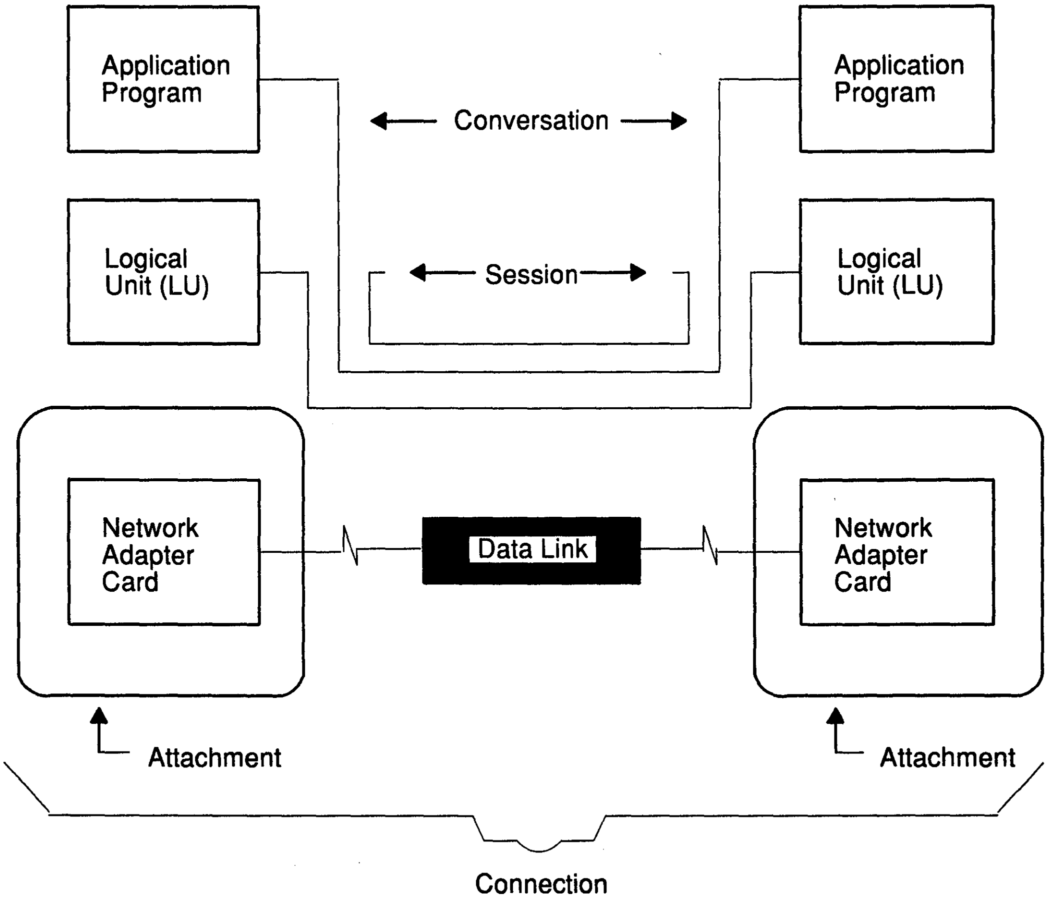


Figure 6. AIX Data Exchange Environment

Attachment

An *attachment* is the physical hardware (adapter and cabling) and supporting programs that allow the AIX Operating System to attach to a particular communications link. The attachment includes the remote node but only describes the physical environment of the node. For AIX SNA Services/6000, the attachment may be one of the following:

- Wide Area Network (WAN)
  - EIA232D SDLC
  - EIA422A SDLC
  - Smart Modem SDLC
- Local Area Network (LAN)
  - Token-Ring
  - Standard Ethernet
  - IEEE 802.3 Ethernet
- Data Network
  - CCITT X.25
  - CCITT X.21 SDLC
  - CCITT V.35 SDLC
  - CCITT V.25 bis SDLC.

Each local attachment is designated as either a *call* (an outgoing attachment to a remote station) or a *listen* (an attachment that accepts incoming calls from a remote station). You can also designate an AIX SNA Services/6000 attachment as *autolisten* by selecting that feature in one of the profiles associated with the attachment.

Autolisten automatically starts another listening attachment when the previous one becomes active (begins transmitting or receiving data). Autolisten has different effects, depending on the type of logical link used:

- For a LAN or X.25 packet network, autolisten remains in listen mode for any incoming network calls.
- For a WAN or X.21 data network, autolisten becomes active when the original attachment becomes inactive.

AIX SNA Services/6000 generates unique names for each autolisten attachment created after the original attachment that was started with a reference to a profile name. For example, if you enter the command:

```
startsrc –t attachment –o linka
```

AIX SNA Services/6000 creates another autolisten attachment with the name `linka0001` when the line becomes active.

The name that AIX SNA Services/6000 generates is always the original attachment profile name, expanded to eight characters by adding zeros to the end, with a nonzero number appended to the end. In this example, when attachment `linka0001` becomes active, AIX SNA Services/6000 generates an autolisten attachment with a name of `linka0002`.

To use an attachment, you must describe its characteristics to AIX SNA Services/6000. As an example, see the attachment profile described in Defining SDLC Attachment Characteristics on page 13–56 as well as the associated logical and physical link profiles to describe the characteristics of the attachment. Refer to Understanding Synchronous Data Link Control in *Communications Programming Concepts* for information to define the data link control (DLC) and Communications I/O Subsystem Overview in *Kernel Extensions and Device Support Programming Concepts* for the characteristics of the adapter associated with the attachment. See Generic Data Link Control (GDLC) Environment in *Communications Programming Concepts* for further information concerning the various data link controls and Devices Overview or Devices Overview for System Management in *General Concepts and Procedures* for information on using and managing devices.

Connection

A *connection* is the network path that links two LUs in different nodes together to enable them to establish communications. Besides the attachment, a connection includes network addressing, the name of the remote program, and other descriptive information.

The connection is created by using one of the following:

- the **startsrc** command
- the **open** subroutine
- the **snaopen** subroutine.

When the connection is created by the **open** or **snaopen** subroutine, the system returns a file descriptor that can be used on subsequent conversations to send and receive data from the partner LU. Many cooperating processes can operate with the same connection, using a different conversation for each process.

Similarly, when using LU 1, 2, or 3, one process can operate with many different connections, using the **select** subroutine. The **close** and **snaclose** subroutines close the file descriptor associated with the connection. The connection is ended by a **stopsrc** connection command.

To use a connection, you must describe its characteristics to AIX SNA Services/6000. As an example, see the connection profile described in Defining LU Type 1 (Local Logical Unit and Logical Connection, on page 13–167 and the profiles associated with it to describe the characteristics of a connection.

Session

A *session* is an established communications path between two network addressable units (NAUs) through which programs can communicate with one another. Two NAUs create a session in response to application program requests to open a conversation.

A session is a relatively long-lived resource that can be used by several different conversations in a serial fashion. Only one conversation uses a session at any one time. The session is not visible to the application program and the application program cannot request a specific session.

Characteristics of a session are described by its *mode*. Use the mode profile described in Defining LU 6.2 Mode Session and Mode List Characteristics on page 13–187. to describe the characteristics of a session to AIX SNA Services/6000.

Conversation

A *conversation* is a pathway between two application programs that allows them to transfer information to one another. The LU selects a session for the conversation and coordinates the interaction between the programs by distinguishing at all times the **send** program from the **receive** program.

The application starts a conversation when it uses one of the following:

- the **ioctl**(ALLOCATE) subroutine
- the **snalloc** subroutine
- the **writex** subroutine (with the allocate bit specified in the **ext_io_str** structure).

A *resource identifier* (rid), returned from the allocation process, identifies the conversation for use in reading and writing data. To end the conversation, use one of the following:

- the **ioctl**(DEALLOCATE) subroutine
- the **snadeal** subroutine
- the **writex** subroutine (with the deallocate bit specified in the **ext_io_str** structure)
- the **close** subroutine.

You define the characteristics of a conversation by defining the characteristics of the two programs involved in the conversation. With AIX SNA Services/6000, you define those characteristics in the TPN profile (see Defining LU 6.2 Transaction Program Name Characteristics on page 13–195) and the RTPN profile (see Defining LU 6.2 Remote TPN Characteristics on page 13–204).

AIX SNA Services/6000 Physical Connections

AIX SNA Services/6000 includes support for the following types of physical network protocols. Once the network is defined using the profiles (see Defining a Network to AIX SNA Services/6000 (SMIT Method) on page 13–50), the type of physical network being used is invisible to users and application programs. However, AIX SNA Services/6000 requires at least one of these types of physical networks:

SDLC

SDLC (Synchronous Data Link Control) is a synchronous data communications protocol used for teleprocessing connections to other systems that support this protocol. Data link control and device driver software are required to use SDLC. A separate 4-Port Multiprotocol Communications Controller and other external hardware are also required. SDLC can be used on the following supported physical links:

- A Standard EIA232D link between two systems (connected either directly or over a public telephone network).
- A Smart Modem link between two systems (connected either directly or over a public telephone network).
- A CCITT recommendation X.21 data network that uses SDLC as its link protocol.
- A Standard EIA422A link between two systems.
- A CCITT recommendation V.25 bis data network that uses SDLC as its link protocol.
- A CCITT recommendation V.35 data network that uses SDLC as its link protocol.

Standard Ethernet

Standard Ethernet is a baseband CSMA/CD local area network protocol used to interconnect several systems on one network. Data link control and device driver software are required to use Ethernet. A separate Ethernet High-Performance LAN Adapter and other external hardware are also required.

IEEE 802.3 Ethernet

IEEE 802.3 Ethernet is a baseband CSMA/CD local area network protocol compatible with the Institute of Electrical and Electronics Engineers Project 802. This protocol includes the IEEE Standard 802.3 common logical link control and the IEEE Standard 802.3 medium access control protocols. Data link control control and device driver software are required to use IEEE Standard 802.3 Ethernet. A separate IEEE Standard 802.3 Ethernet High-Performance LAN Adapter and other external hardware are also required.

Token-Ring

Token-Ring is a baseband local area network configured into a star-wired ring and specifically designed to be used with the IBM Cabling System. Multiple rings can be connected using IBM Token-Ring Network Bridges. Data link control and device driver software are required to use Token-Ring. A separate Token-Ring High-Performance Network Adapter and other external hardware are also required.

X.25

X.25 is a packet-switched data network protocol compatible with the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation X.25. It includes the IBM Qualified Logical Link Control (QLLC) protocol and the CCITT X.25 medium access control. Data link control and device driver software are required to use X.25. A separate X.25 High-Performance Network Adapter and other external hardware are also required.

EIA232D Physical Link Control

The EIA232D physical link control supports the logical interface defined by the Electronics Industries Association (EIA) Standard EIA232D and the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation V.24, with the following optional features:

- Data rates of up to and including 19.2K bps per physical port
- Direct connections of up to 50 feet between stations
- Switched and nonswitched DCE (data circuit-terminating equipment) connections
- Manual call
- Manual and automatic answer
- Full and alternate speed
- Business machine clocking and external bit clocking
- Controlled and continuous request to send (RTS)
- Call override of listen attachments
- Automatic restart of listen attachments.

Smart Modem Physical Link Control

The Smart Modem physical link control supports the logical interface defined by the Electronics Industries Association (EIA) Standard EIA232D and the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation V.24, with the following optional features:

- Data rates of up to and including 19.2K bps per physical port
- 79-character ASCII modem command sequence buffer
- Switched and nonswitched DCE connections
- Manual and automatic call
- Manual and automatic answer
- Full and alternate speed
- Controlled and continuous RTS
- Call override of listen attachments
- Automatic restart of listen attachments.

X.21 Physical Link Control

The X.21 physical link control supports the general purpose interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for synchronous operation on a public data network (PDN), as defined by the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation X.21. The following optional features are supported:

- Data rates of up to and including 64K bps per physical port
- 255-character IA5 call-selection sequence buffer
- Leased or switched circuits
- Direct or selected calls
- Abbreviated address calling
- Facilities requests
- Called and calling line identification
- Japanese Shift-In start delimiter
- Automatic call retries on specific call progress signals
- Call override of listen attachments
- Automatic restart of listen attachments.

EIA422A Physical Link Control – Long Distance Drivers

The EIA422A physical link control supports the logical interface defined by the Electronics Industries Association (EIA) Standard EIA422A and the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation V.24, with the following optional features:

- Data rates of up to and including 19.2K bps per physical port
- Direct connections greater than 50 feet between stations
- Full and alternate speed
- Internal and external bit clocking (External to 64K, internal to 38K. Does not provide clock signal.)

- Controlled and continuous RTS
- Call override of listen attachments
- Automatic restart of listen attachments.
- Leased line support.

V.25 bis Physical Link Control

The V.25 bis physical link control supports the logical interface defined by the Electronics Industries Association (EIA) Standard EIA232D and the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation V.25 bis, with the following optional features:

- Data rates of up to and including 19.2K bps per physical port
- 79-character ASCII modem command sequence buffer
- Switched and nonswitched DCE connections
- Manual and automatic call
- Manual and automatic answer
- Full and alternate speed
- Controlled and continuous RTS
- Call override of listen attachments
- Automatic restart of listen attachments.

V.35 Physical Link Control

The V.35 physical link control supports the logical interface defined by the Electronics Industries Association (EIA) Standard EIA422A and the International Telegraph and Telephone Consultative Committee (CCITT) Recommendation V.35, with the following optional features:

- Data rates of up to and including 19.2K bps per physical port
- Direct connections greater than 50 feet between stations
- Switched and nonswitched DCE (data circuit-terminating equipment) connections
- Manual call
- Manual and automatic answer
- Full and alternate speed
- Internal and external bit clocking (External to 64K, internal to 38K. Does not provide clock signal.)
- Controlled and continuous RTS
- Call override of listen attachments
- Automatic restart of listen attachments
- Leased line support.

AIX SNA Services/6000 Logical Connections

Using any of the supported physical networks (see AIX SNA Services/6000 Physical Connections on page 13–14), you can use AIX SNA Services/6000 logical unit (LU) protocols to connect an AIX node to the following categories of remote systems:

- Host systems (Ethernet is not currently available on supported host systems)
- Peer systems.

Host Systems

A host system starts and controls all exchanges between it and any attached subordinate systems. The host system is usually a larger computer system, with the capability to be an SNA T4 or T5 subarea node. The AIX Operating System workstation cannot be a host system. To communicate with a host system, the AIX workstation can use one of the following LU protocols provided by AIX SNA Services/6000:

- LU0
- LU 1
- LU 2
- LU 3
- LU 6.2 operating as a dependent LU in migration mode.

In a host to workstation configuration, the AIX workstation operates like a remote terminal (T2.1 boundary function) with respect to the host system. It depends on the host system for control of the sessions between the two systems. The following figure illustrates this AIX workstation to host connection.

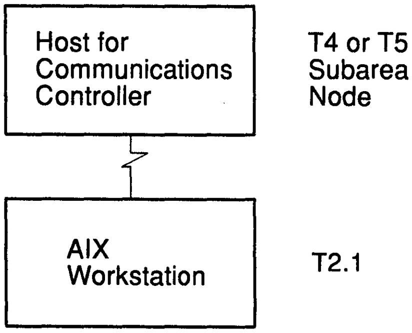


Figure 7. AIX to Host Connection

Peer Systems

A peer system operates like an equal to the other systems on the network with which it communicates. Both systems involved in data communication cooperate in establishing communications between the two systems. This type of communications is supported only by LU 6.2. Each system in the network operates as a T2.1 peripheral node. Using the LU 6.2 protocol, the AIX Operating System can communicate with other AIX systems or with other computer systems that support the LU 6.2 protocol in peer-to-peer communications.

The following figure illustrates a typical local area network configuration of AIX workstations connected in peer-to-peer communications.

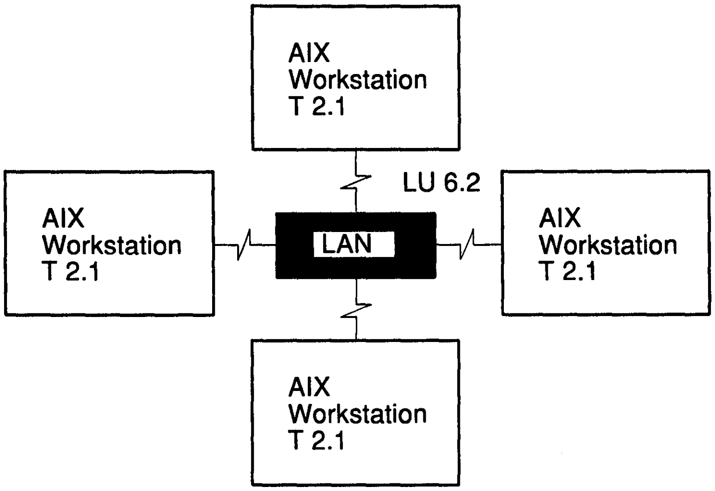


Figure 8. AIX Peer-to-Peer Communications on a Local Area Network

Connecting Two Different Networks

A good example of interconnected systems is that of an AIX workstation connected to other AIX workstations and a host system. One port on the AIX workstation connects to a local area network, using LU 6.2 protocols. Another port connects to a host system, using LU 6.2, LU 0, LU 1, LU 2, or LU 3 protocols.

In this configuration, the AIX workstation operates as both a member of the local area network and as a host-connected node. Each operational mode is independent from the other. This configuration does *not* allow another member of the local area network to pass through the AIX workstation to communicate with the host system. The following figure shows an AIX workstation connected to other AIX workstations and a host system, illustrating the previous explanation of how to interconnect systems.

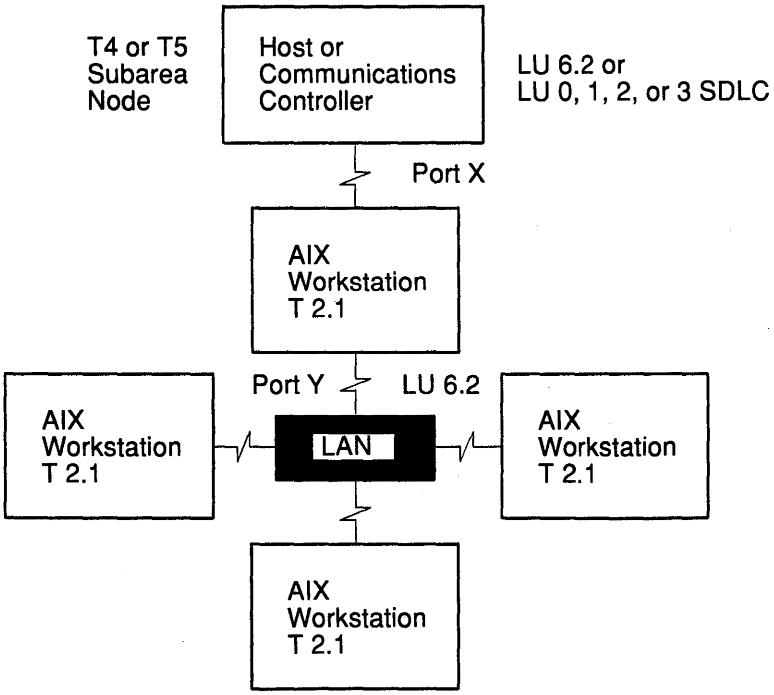


Figure 9. AIX Workstation Connected to Two Networks

How AIX SNA Services/6000 Stores Information

AIX SNA Services/6000 keeps all network information in structured files called *profiles.* A profile is a structured collection of data that describes the significant features of a user, program, or device. AIX SNA Services/6000 stores the profiles as database files that are controlled by system data management routines. All AIX SNA Services/6000 components can access the information from the profiles by requesting the information from the data management routine. An explanation of the different components responsible for SNA Services/6000 information storage follows the following figure illustrating how AIX SNA Services/6000 stores information.

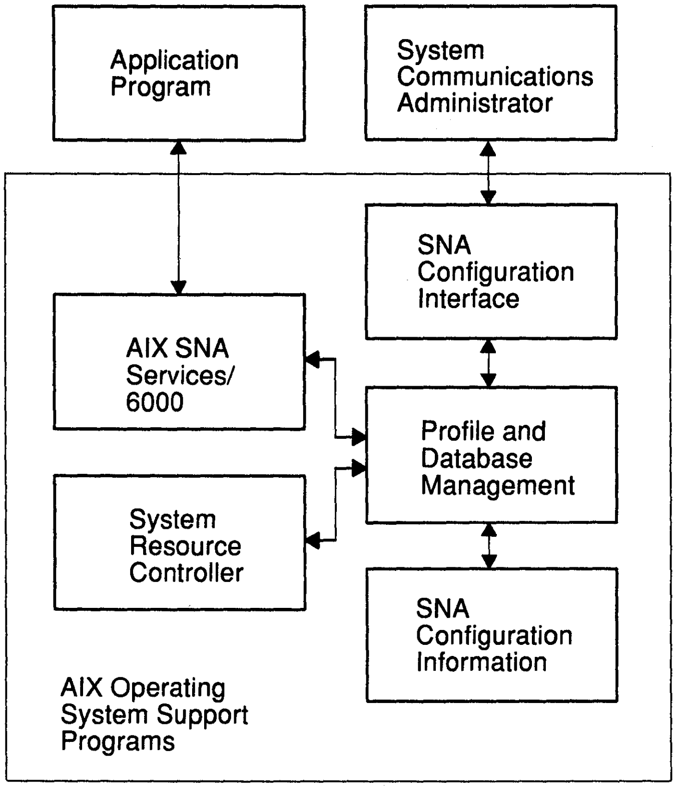

Application Program
System Communications Administrator
SNA Configuration Interface
AIX SNA Services/ 6000
Profile and Database Management
System Resource Controller
SNA Configuration Information
AIX Operating System Support Programs

Figure 10. AIX SNA Services/6000 Information Storage

SNA Configuration Interface

A set of configuration commands is provided with AIX SNA Services/6000 to allow you to set up your network. These commands are accessed directly from the AIX Operating System command line or through the SMIT Interface. Refer to Defining a Network to AIX SNA Services/6000 (System Management Interface Tool (SMIT) Method) on page 13–50 for more information about using the SMIT Interface. Refer to AIX SNA Services/6000 Commands on page 13–256 for more information about using the configuration commands directly from the AIX Operating System command line.

Database Management

A set of database management services is provided by the AIX Operating System that allows the SNA configuration commands to access the database. When profiles are stored in the database, they are stored in a different format than regular AIX Operating System files for easier information retrieval. Therefore, when information is retrieved from the database, the requesting component must format the data for presentation to the user.

SNA Configuration Information

The configuration information is stored in structured profiles. These profiles can only be read through subroutine calls to the Profile and Database Management component by other system components. You cannot browse, edit, or copy the files, using ordinary AIX Operating System commands or utilities. Backing Up SNA Configuration Profiles on page 13–47 explains how to back up information in these files, using AIX Operating System commands. Customizing AIX SNA Services/6000, on page 13–45, explains how to use the SMIT Interface to manipulate the information in these profiles.

The figure on page 13–23 shows a hierarchical description of the following AIX SNA Services/6000 profiles. These profiles are described more completely in the following pages. The profiles and their functions are as follows:

Profile **Function**

SNA Describes the characteristics of the SNA background program and its environment for use by the System Resource Controller (SRC). Refer to the System Resource Controller Overview in *General Concepts and Procedures* for more information on the SRC.

Connection Together with the profiles below it in the hierarchy, describes the characteristics of a connection to a system. One connection profile exists to describe each LU-Partner pair.

Attachment Together with the profiles below it in the hierarchy, describes the characteristics of the network link. The attachment profile can be used with many connection profiles. It includes pointers to the following profiles:

- Control Point
- Logical Link
- Physical Link.

Control Point Describes the characteristics of the physical unit (PU) associated with the local system. The same control point profile can be used with many attachment profiles.

Logical Link Describes the characteristics of the link protocol used on the network. Only one logical link protocol exists for an attachment. It can be one of the following:

- SDLC (Synchronous Data Link Control)
- Standard Ethernet
- Token-Ring
- IEEE 802.3 Ethernet
- QLLC.

The same logical link profile can be used with many attachment profiles.

LU Address Registration
Lists the LU addresses that are reserved for generic SNA sessions.

Physical Link Describes the characteristics of the physical link provided by the network. The physical link can be one of the following:

- Standard Ethernet
- Token-Ring
- IEEE 802.3 Ethernet
- X.25 (QLLC only)
- Smart Modem (SDLC only)
- EIA232D (SDLC only)
- EIA422A (SDLC only)
- X.21 (SDLC only)
- V.25 bis (SDLC only)
- V.35 (SDLC only).

The same physical link profile can be used with many attachment profiles.

Local LU Describes the characteristics of the local LU (logical unit). It contains a pointer to the transaction program name (TPN) list profile. The same local LU profile can be used with many connection profiles.

TPN List A list of all transaction program names (TPNs) that can run on the associated connection. A TPN is an application program that runs on the local system and communicates on the network, using AIX SNA Services/6000. The TPN list profile can be used with many different local profiles.

TPN Describes the characteristics of one of the application programs on the local system that can run on the associated connection. This profile contains a pointer to the actual transaction program to allow the program to be started on the connection. The same TPN profile can appear in many TPN list profiles, allowing the same application program to run on many different connections.

RTPN List A list of all remote transaction program names (RTPNs) that can run on the associated connection. An RTPN is an application program that runs on the remote system and communicates on the network, using AIX SNA Services/6000. The same RTPN list profile can be used with many different connection profiles.

RTPN Describes the characteristics of one of the application programs on the remote system that runs on the associated connection. This profile contains a pointer to the actual transaction program to allow the program to be started on the connection.

When using LU 6.2 peer-to-peer communications, AIX SNA Services/6000 automatically starts the remote TPN on the remote node when the conversation is started. The same RTPN profile can appear in many RTPN list profiles, allowing one remote program to run on many different connections.

Mode List A list of all mode profiles that can be used to describe the rules for the associated connection. The same mode list profile can be used with many different connections.

Mode Describes one of the sets of rules that can govern the operation of the associated connection. The same mode profile can appear in many mode list profiles, thus governing the operation of many different connections.

The figure below shows a hierarchical description of the aforementioned AIX SNA Services/6000 profiles.

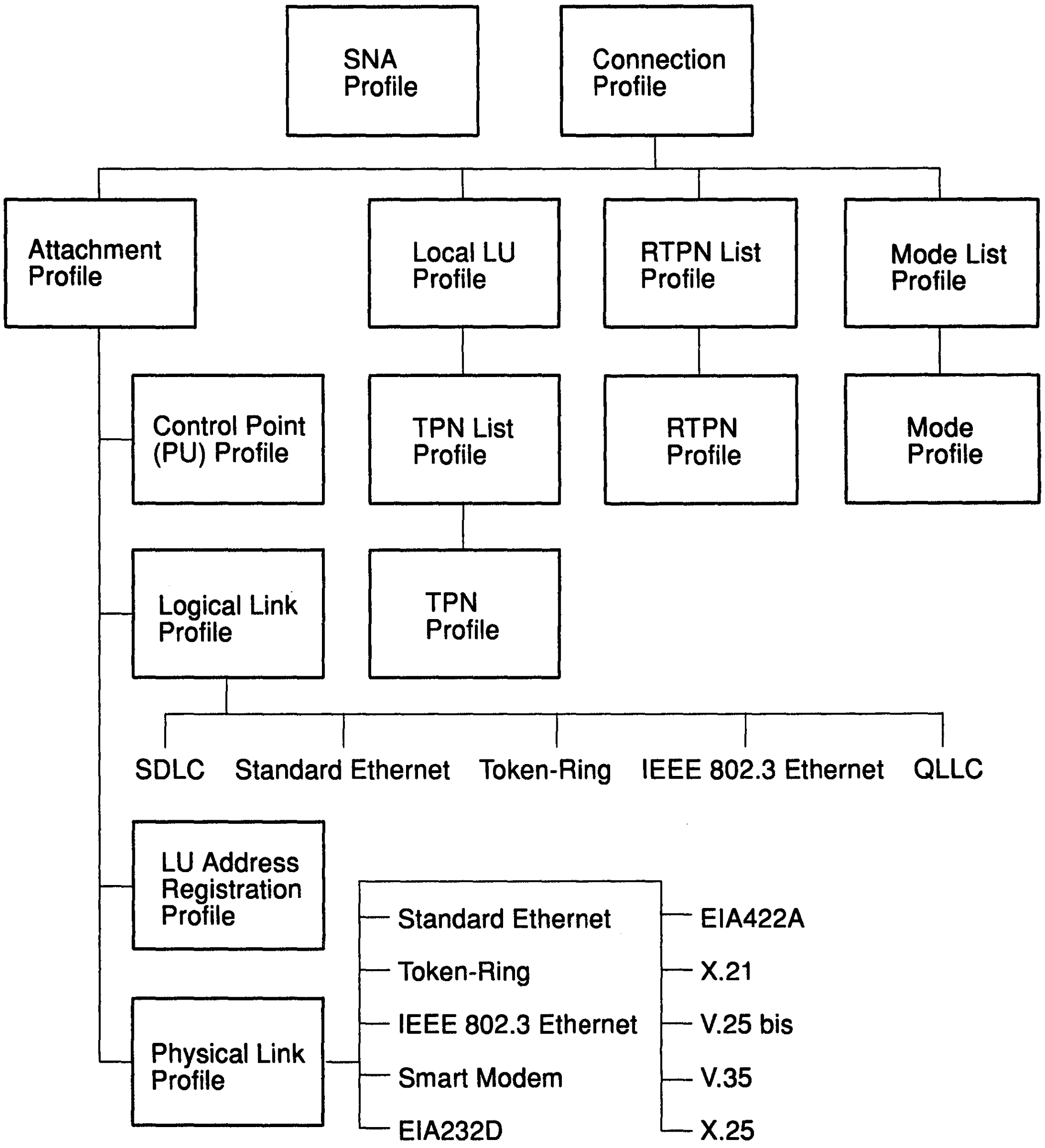


Figure 11. Profile Name Links

AIX SNA Services/6000

AIX SNA Services/6000 is a set of background processes, AIX kernel processes, and AIX Operating System device drivers that run under the control of the System Resource Controller (SRC). The System Resource Controller, in turn, provides all services to transfer data over the network. To provide these services, the SRC requests information from the Profile and Data Management components to define the characteristics of connections and attachments on the network. Refer to the System Resource Controller Overview in *General Concepts and Procedures* for more information on the SRC.

The following background processes run as part of AIX SNA Services/6000:

luxcr Performs a command-routing function and provides a common interface for the other background processes to the System Resource Controller.

luxcps Provides Control Point Services to start and stop attachments to other nodes in the network and informs the other background processes of exception conditions related to network attachments.

luxlrm Performs a Logical Unit Resource Management function.

luxlns Performs a Logical Unit Network Services function.

luxihd Performs an interrupt-handling function, providing an interface for the AIX SNA Services/6000 background processes to the AIX kernel processes.

These processes must be running in order for AIX SNA Services/6000 to function properly. The **luxcr** process is started by the System Resource Controller. The **luxcr** process, in turn, starts the other processes.

The following AIX Operating System Device Drivers are part of AIX SNA Services/6000:

- SNA Data Device Driver
- SNA Manager Device Driver
- SNA Generic Device Driver.

These Device Drivers must be present for AIX SNA Services/6000 to function properly. They should appear in a listing of the /**dev** directory as /**dev/sna**, /**dev/snam**, and /**dev/gsna** respectively. The SNA Data Device Driver provides the AIX Operating System subroutine interface to AIX SNA Services/6000 defined later in this document.

The SNA Manager Device Driver provides an interface for the AIX SNA Services/6000 background processes to AIX Operating System subroutines that implement control functions internal to AIX SNA Services/6000. The SNA Manager Device Driver interface is for the exclusive use of the AIX SNA Services/6000 background processes.

The SNA Generic Device Driver provides a PIU level subroutine interface for generic LU application. The LU 0 support provided with AIX SNA Services/6000 is written to the generic device driver interface.

AIX SNA Services/6000 Network Names and Addresses

AIX SNA Services/6000 uses many different identifiers to ensure that information is sent to the proper destination. Each identifier, in the form of a name or address, applies to a specific part of the network. Some identifiers refer to physical elements while others refer to logical elements. Although not all identifiers are required, each plays a part in routing information. The following paragraphs discuss the identifiers used in typical network configurations and indicate what each identifier means and where you must enter that identifier into the system.

Note: In the following discussions, an *optional* identifier means that the name is not required to make the network operate. If you do not supply the identifier, AIX SNA Services/6000 makes the connection or session with the most logical party. For example, if the system is linked to only one other logical unit, you should not need to specify a remote LU name to reach that LU. When you do not specify an optional identifier, you must be willing to accept the value that AIX SNA Services/6000 provides.

AIX SNA Services/6000 Naming Requirements

When describing a connection to AIX SNA Services/6000, you must provide names for profiles, network identifiers, and other parameters. Depending upon where it is used, the name must conform to different rules regarding the characters that can be used to form the name. These rules, by category, are as follows:

- **AIX Operating System File Names**
  These can be any names that are legal for AIX Operating System file names. Names must begin with either a letter or number, and the characters * and / are not allowed.
- **Character Set A**
  This character set is a limited set of characters used for external network identifiers. The character set consists of:
  - Uppercase letters
  - Numbers 0 to 9
  - Special characters $, #, and @.

  The first character of a name using this character set must be an uppercase letter or a special character.
- **Character Set AE**
  This character set is an extended version of Character Set A. It is also used for external network identifiers. The character set consists of:
  - Uppercase and lowercase letters
  - Numbers 0 to 9
  - Special characters $, #, and @
  - A . (period).

  There is no restriction on the first character of a name using this character set.

The following tables list the names required for AIX SNA Services/6000, the length of each name, and the restrictions that apply to particular names.

Profile Name Links		Part 1 of 5
Name	**Size/Type**	**Restrictions**
Device Name	1–15 ASCII characters	Must match a device name defined to the system.

Profile Name Links		Part 2 of 5
Network-Related Name	**ASCII characters**	**Character set A**
Local Link Name	1–8	Yes
Local LU Name	1–8	Yes
Mode Name	1–8	Yes
Network Name	1–8	Yes
Remote Link Name	1–8	Yes
Remote LU Name	1–8	Yes

Profile Name Links		Part 3 of 5
Profile Name	**ASCII characters**	**AIX Operating System file names**
Attachment	1–8	Yes
Connection	1–14	Yes
Control Point	1–14	Yes
Local LU	1–14	Yes
Local Link	1–8	Yes
Mode	1–14	Yes
Mode List	1–14	Yes

Profile Name Links		Part 4 of 5
Name	**Size/Type**	**Restrictions**
Physical Link	1–14	None
Remote TPN	1–14	None
Remote TPN List	1–14	None
SNA	1–14	Cannot begin with a number.
TPN	1–14	Cannot begin with a number.
TPN List	1–14	None

Profile Name Links **Part 5 of 5**

Program Name	Size/Type	Restrictions
Remote TPN:		
Character	1–64	Character set AE
Hexadecimal	1–128	Any hexadecimal digits 0–9, A–F.
TPN:		
Character	1–64	Character set AE
Hexadecimal	1–128	Any hexadecimal digits 0–9, A–F.
XID Node ID	8 hexadecimal digits	3-digit block number; 5-digit sequence number.
X.21 Selection Sequence	1–255 decimal digits and the *, +, ,, ., –, and /.	A + (LA5 character set) is added to the end as a terminator.
V.25 Modem Command Sequence	1–79 ASCII characters	A carriage return (0X0D) is added to the end as a terminator.
Smart Modem Command Sequence	1–79 ASCII characters	A carriage return (0X0D) is added to the end as a terminator.
Remote Station X.25 Address	1–15 ASCII characters	Digits 0–9

LU 6.2 Network Names

Following is a description of two possible peer-to-peer connections, delineating the optional and required AIX SNA Services/6000 identifiers described in the following pages. The first example describes an Ethernet connection and the second an SDLC connection.

The Ethernet connection in an LU 6.2 peer-to-peer configuration requires only the Remote Link Name on the remote system. The local system requires the Data Link Device Name, the Local SAP Address, and the Local Link Name.

Optional identifiers for the remote system are the Mode, the Remote LU Name, and the Remote Network Name. Optional identifiers for the local system are the Mode, the Local LU Name, and the Local Network Name.

An SDLC connection in an LU 6.2 peer-to-peer configuration has significantly different requirements from the Ethernet connection. The remote system has no required identifiers. The local system requires only the Data Link Device Name.

Optional identifiers for the remote system are the Mode, the Remote LU Name, the Remote Network Name, and the Remote Secondary Station Address. Optional identifiers for the local system are the Mode, the Local LU Name, the Local Network Name, and the Local Secondary Station Address.

The figure on page 13–28 illustrates the previous explanation of required and optional AIX SNA Services/6000 identifiers. The figure shows both the required and optional identifiers for each configuration as well as the location of the logical element associated with each identifier, remote or local.

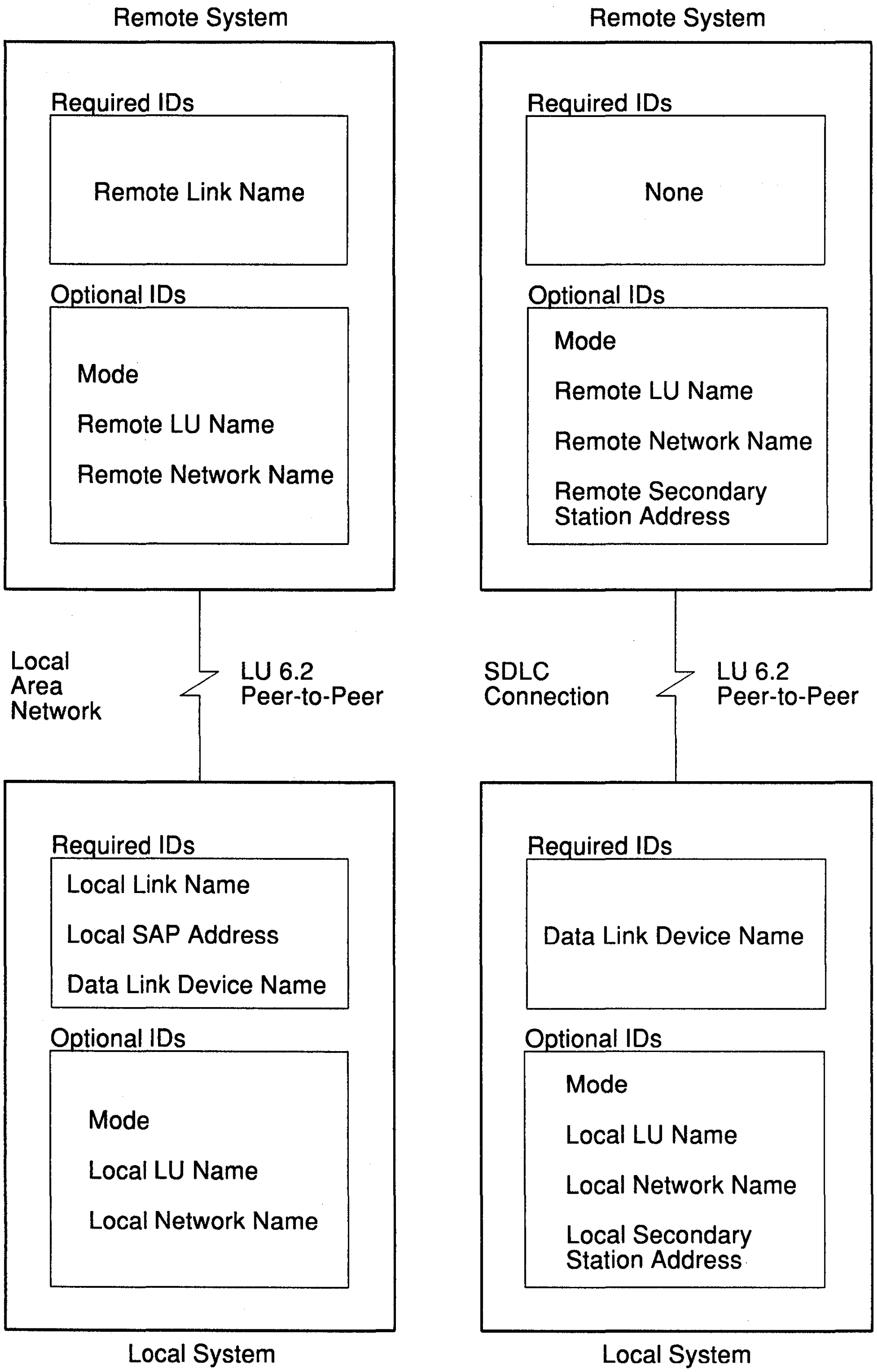


Figure 12. LU 6.2 Network Configurations

AIX SNA Services/6000 Identifiers

The following identifiers are used in most AIX SNA Services/6000 configurations Each of these identifiers is fully described in the description of the profile in which it is used. The following paragraphs summarize their usage:

Identifier **Use**

Mode
Identifies the name of the mode that describes the characteristics of a session. This name must be the same for both the remote and local systems using that session. Enter the mode name in the mode profile.

Local LU Name
Identifies the local LU involved in the session. This name is assigned when setting up the network. Enter this name in the local LU profile. The local LU name on the local system is the remote LU name on the remote system.

Remote LU Name
Identifies the remote LU involved in the session. This name is assigned when setting up the network. Enter this name in the connection profile. The remote LU name on the local system is the local LU name on the remote system.

Network Name (Local)
Identifies the name that the network uses to identify the local node. This name is assigned when the node is added to the network. The local network name on the local system is the remote network name on the remote system. Enter the local network name in the local LU profile.

Network Name (Remote)
Identifies the name that the network uses to identify the remote node. This name is assigned when the node is added to the network. The remote network name on the local system is the local network name on the remote system. Enter the remote network name in the connection profile.

LAN Identifiers

Identifier **Use**

The following identifiers are used in most AIX SNA Services/6000 physical link configurations Each of these identifiers is fully described in the description of the profile in which it is used. The following paragraphs summarize their usage:

Remote SAP Address
Specifies the Service Access Point address of the remote system entered in the logical link profile of the local system. This address is used and defined by the network but does not have to be unique within the network. The remote SAP address on the local system is the local SAP address on the remote system.

Local SAP Address
Specifies the Service Access Point address of the local system that is entered in the logical link profile of the local system. This address is used and defined by the network but does not have to be unique within the network. The local SAP address on the local system is the remote SAP address on the remote system.

Link Name (Local)
Identifies the local node to other nodes on the network. This name cannot be assigned to any other node on the network. For this parameter, select any string of characters that has meaning to you.

Link Name (Remote)

Identifies the remote node to other nodes on the network. This name cannot be assigned to any other node on the network. This name must match the local link name entered at the remote system.

Data Link Device Name

The AIX Operating System device name of the data link device manager as defined to the system. Enter this name in the applicable physical link profile. The local AIX SNA Services/6000 uses this name both to attach itself and interface to the data link control (DLC).

SDLC Identifiers

The following identifiers are used in AIX SNA Services/6000 SDLC logical and physical link configurations Each of these identifiers is fully described in the description of the profile in which it is used. The following paragraphs summarize their usage:

Identifier **Use**

Secondary Station Address

Identifies the local or remote node to the SDLC primary system. This address is defined by the host system.

Data Link Device Name

The AIX Operating System device name of the network adapter. Enter this name in the applicable physical link profile. The local AIX SNA Services/6000 uses this name to get information that defines the interface to the network adapter.

Host Attachment Identifiers

When attached to a remote host computer, the AIX node always operates as a secondary station. An SDLC connection in an LU 6.2 peer-to-peer configuration requires the following identifiers explained below. The host system (primary) requires the SSCP ID and XID Node ID. The local AIX system (secondary) requires the Data Link Device Name, the LU Address, and the Local Secondary Station Address.

Optional identifiers for the host system are the Mode, the Remote LU Name, and the Remote Network Name. Optional identifiers for the local AIX system are the Mode, the Local LU Name, and the Local Network Name. An SDLC connection in an LU 0, 1, 2, or 3 configuration has the same identifiers as the LU 6.2 configuration with the exception of the Mode identifier, which is not required.

The figure on page 13–31 illustrates these two configurations using LU 6.2 and LU 0, 1, 2, or 3. The following identifiers are unique to these configurations:

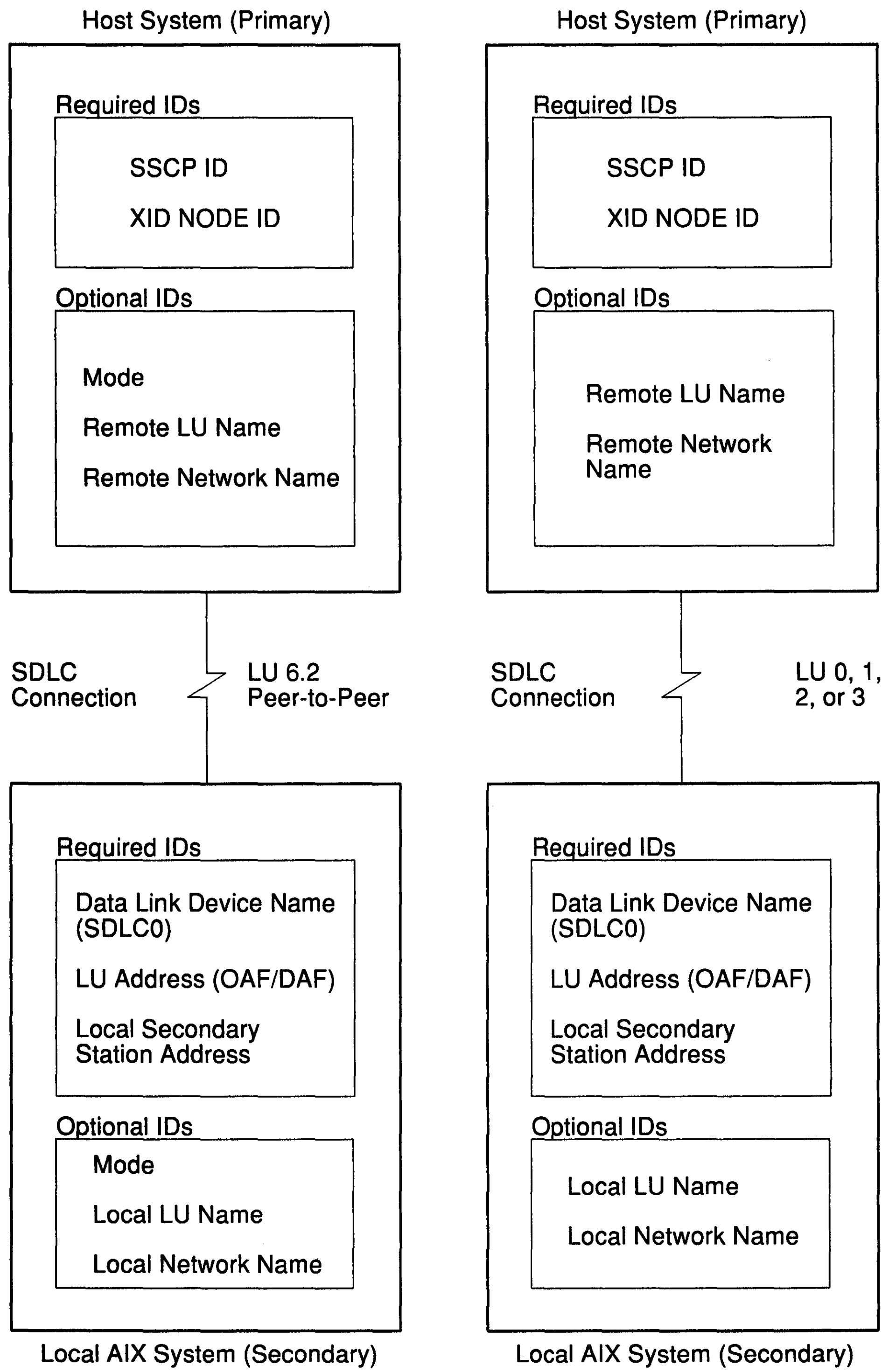


Figure 13. Host Attachment Configurations

Identifier **Use**

SSCP ID The remote control point ID exchanged during the ACTPU from the host. This identifier is defined by the host system. Enter this identifier in the local LU profile.

XID Node ID Identifies the host system at the start of a session with the secondary station. This name is defined when the host system is generated. Enter this identifier in the control point profile.

Local Secondary Station Address

Identifies the local node to the SDLC host system. This address is defined by the host system.

LU Address (Local)

Assigned by the network to identify the local LU to the network. This address is used as the DAF (destination address field) in the transmission header. Enter this address in the local LU profile and in the LU Address Registration for LU 0 support.

LU 6.2 Sample Network with Profiles

The following example depicts a sample network that connects a department system in a small college with the college's central computer. The department system at node 1 (CS_DEPT) belongs to the Computer Science department. It is connected through an LU 6.2 link to the central computer at node 2 (MASTER). The Computer Science department has three connections to the central computer each defined for different purposes on that system:

- Connection_A
- Connection_B
- Connection_C.

Note: Although host computers do not use the AIX Operating System concepts of *attachment*, *connection*, and *configuration profiles*, for illustration purposes the following examples express the host configurations in terms of what the equivalent profile setup would be like.

The central computer has the following three corresponding connections defined to handle the requirements of the Computer Science department:

- Connection_X
- Connection_Y
- Connection_Z.

The following figure illustrates this sample network connecting the department system in a small college with the college's central computer.

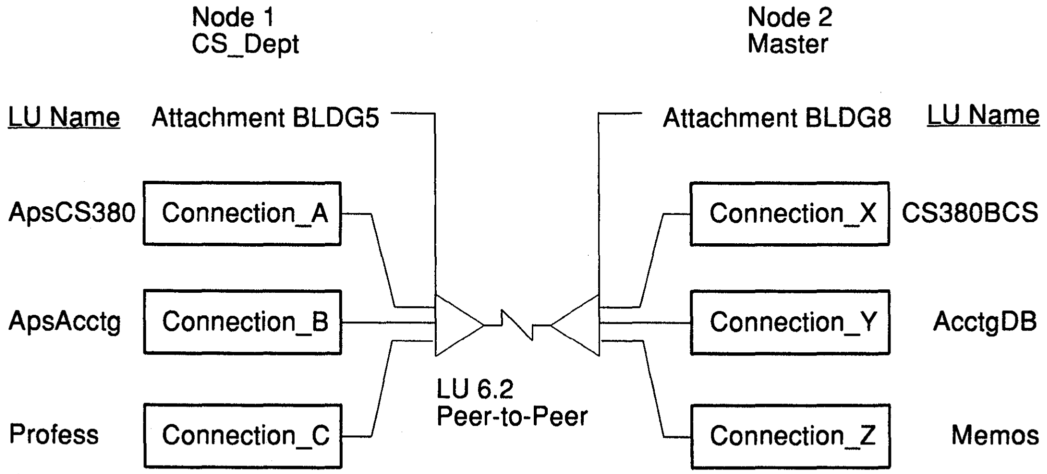


Figure 14. Sample SNA Network

Connection_A and Connection_X

The Computer Science department set up Connection_A to allow students from a certain class to access the central computer for class exercises and research. The staff at the central computer center set up a complementary connection, Connection_X, to handle their end of the link. The following table shows how each group set up the names in their respective profiles to establish the link. Notice the following points:

- The Local LU Name field for Node 1 matches the Remote LU Name field for Node 2.
- The Network Name field of the local LU profile for Node 1 matches the Network Name field of the connection profile for Node 2.
- Both connections use the same Mode Name field.
- The Network Name field of the connection profile for Node 1 matches the Network Name field of the local LU profile for Node 2.
- The Remote LU Name field for Node 1 matches the Local LU Name field for Node 2.

Names in Profiles for Connections A and X

Profile Type	Field	Node 1 Content	Node 2 Content
Connection	Connection Profile Name	Connection_A	Connection_X
	Attachment Profile Name	BLDG8	BLDG5
	Network Name	MASTER	CS_DEPT
	Remote LU Name	CS380BCS	APSCS380
Local LU	Local LU Name	APSCS380	CS380BCS
	Network Name	CS_DEPT	MASTER
	Independent LU	Yes	Yes
Mode	Mode Name	STUDENTS	STUDENTS
Attachment	Attachment Profile Name	BLDG8	BLDG5
	Logical Link Type	SDLC	SDLC
Logical Link	Logical Link Device Name	SDLLC0	SDLLC0
Physical Link	Data Link Device Name	SDLC0	SDLC0

Connection_B and Connection_Y

The Computer Science department set up Connection_B to handle departmental accounting and to gain access to sensitive personnel data. The staff at the central computer center set up a complementary connection, Connection_Y, to handle their end of the link.

The following table shows how each group set up the names in their respective profiles to establish the link. Notice that the similarities noted in Connection_A and Connection_X on page 13–33 apply here as well.

Names in Profiles for Connections B and Y			
Profile Type	**Field**	**Node 1 Content**	**Node 2 Content**
Connection	Connection Profile Name	Connection_B	Connection_Y
	Attachment Profile Name	BLDG8	BLDG5
	Network Name	MASTER	CS_DEPT
	Remote LU Name	ACCTGDB	APSACCTG
Local LU	Local LU Name	APSACCTG	ACCTGDB
	Network Name	CS_DEPT	MASTER
	Independent LU	Yes	Yes
Mode	Mode Name	SECURE1	SECURE1
Attachment	Attachment Profile Name	BLDG8	BLDG5
	Logical Link Type	SDLC	SDLC
Logical Link	Logical Link Device Name	SDLLC0	SDLLC0
Physical Link	Data Link Device Name	SDLC0	SDLC0

Connection_C and Connection_Z

The Computer Science department set up Connection_C for use by the professors in the department to handle student grading and official correspondence. The staff at the central computer center set up a complementary connection, Connection_Z, to handle their end of the link. The following table shows how each group set up the names in their respective profiles to establish the link. Notice that the same similarities noted in Connection_A and Connection_X on page 13–33 apply here as well.

Names in Profiles for Connections C and Z			
Profile Type	**Field**	**Node 1 Content**	**Node 2 Content**
Connection	Connection Profile Name	Connection_C	Connection_Z
	Attachment Profile Name	BLDG8	BLDG5
	Network Name	MASTER	CS_DEPT
	Remote LU Name	MEMOS	PROFESS
Local LU	Local LU Name	PROFESS	MEMOS
	Network Name	CS_DEPT	MASTER
	Independent LU	Yes	Yes
Mode	Mode Name	SECURE2	SECURE2
Attachment	Attachment Profile Name	BLDG8	BLDG5
	Logical Link Type	SDLC	SDLC
Logical Link	Logical Link Device Name	SDLLC0	SDLLC0
Physical Link	Data Link Device Name	SDLC0	SDLC0

LUs 1, 2, 3 Sample Network with Profiles

The Computer Science department also decided to implement an LU 1, 2, 3 connection to the central computer from their AIX node. Using this connection, they could install a program that emulates an IBM 3270 data terminal to provide access to the central computer from three terminals on the AIX node. The following figure shows a representation of that connection.

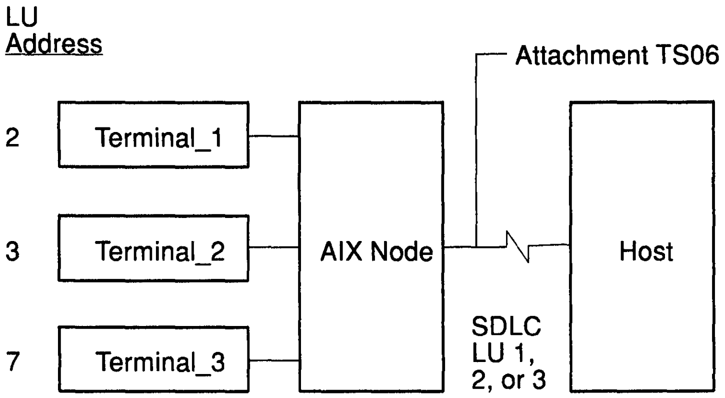


Figure 15. Sample SNA Network

The following table shows the names entered in the profiles on the AIX node for each of the three connections:

Names in Profiles for Sample Host Attachment				
Profile Type	**Field**	**Terminal_1**	**Terminal_2**	**Terminal_3**
Connection	Connection Profile Name	Terminal_1	Terminal_2	Terminal_3
	Attachment Profile Name	TS06	TS06	TS06
Local LU	Local LU Address	2	3	7
	SSCP ID	0500000000029	0500000000029	0500000000029
Control Point	XID Node ID	05C005001	05C005001	05C005001
Attachment	Attachment Profile Name	TS06	TS06	TS06
	Logical Link Type	SDLC	SDLC	SDLC
SDLC Logical Link	Local Secondary Station Address	6	6	6
Logical Link	Logical Link Device Name	SDLLC0	SDLLC0	SDLLC0
Physical Link	Data Link Device Name	SDLC0	SDLC0	SDLC0

AIX SNA Services/6000 LU 6.2

LU 6.2 (logical unit type 6.2) provides a connection between its transaction programs and network resources. The resources may be *local* (connected to this logical unit) or *remote* (connected to another logical unit). In particular, LU 6.2 provides a connection between a local transaction program and a remote transaction program. This interprogram communication on a peer-to-peer basis is called *APPC* (Advanced Program-to-Program Communication).

Interprogram communication allows two transaction programs on separate LUs to interact with each other over the network without being involved in the control of the network. To the transaction programs involved, the connection appears to be a direct connection between the two programs except for the delays involved in transmitting over the network. AIX SNA Services/6000 LU 6.2 provides a uniform programming interface for the transaction programs, buffers the data, and then transmits that data over the network to the other transaction program.

AIX SNA Services/6000 provides both an AIX Operating System subroutine and an SNA library subroutine interface to LU 6.2 functions for application programs. For example, an application program can access the large list of SNA functions by using a few subroutines:

- the **open** subroutine
- the **readx** (or **read**) subroutine
- **writex** (or **write**) subroutine
- the **close** subroutine
- the **select** subroutine
- the **ioctl** subroutine.

The **readx** and **writex** subroutines are expanded versions of the **read** and **write** subroutines. Their extended structure performs the same function as an **ioctl** subroutine in addition to the read or write function.

LU 6.2 Protocol Boundaries

AIX SNA Services/6000 LU 6.2 supports the following sets of protocol boundaries (verbs) as defined in the *SNA Transaction Programmer's Reference Manual for LU Type 6.2.*

- Conversation protocol boundary (basic conversation verbs). See the next table for a listing of these verbs and how they are implemented in AIX SNA Services/6000.
- End user and program-to-program protocol boundary (mapped conversation verbs). See the table on page 13–38 for a listing of these verbs and how they are implemented in AIX SNA Services/6000.
- Control operator protocol boundary (control operator verbs). These verbs are handled internal to AIX SNA Services/6000.

Refer to AIX SNA Services/6000 Subroutines in *Communications Programming Concepts* for descriptions of the subroutines listed in the tables. These documents contain complete information about the supported functions.

AIX SNA Services/6000 Basic Conversation Verbs		
SNA Verb	**Subroutine**	**Subroutine**
ALLOCATE	**ioctl**(ALLOCATE)	**snalloc**
CONFIRMED	**ioctl**(CONFIRMED)	**snactl**(CONFIRMED)
DEALLOCATE	**ioctl**(DEALLOCATE)	**snadeal**
FLUSH	**ioctl**(FLUSH)	**snactl**(FLUSH)
GET_ATTRIBUTE	**ioctl**(GET_ATTRIBUTE)	**snactl**(GET_ATTRIBUTE)
POST_ON_RECEIPT	Not supported	Not supported
PREPARE_TO_RECEIVE	**ioctl** (PREPARE_TO_RECEIVE)	**snactl** (PREPARE_TO_RECEIVE)
CONFIRM	**ioctl**(CONFIRM)	**snactl**(CONFIRM)
RECEIVE_AND_WAIT	**read, readx**	**snaread**
RECEIVE_IMMEDIATE	**read, readx** (NO_DELAY flag on)	**snaread**(NO_DELAY flag on)
REQUEST_TO_SEND	**ioctl** (REQUEST_TO_SEND)	**snactl**(REQUEST_TO_SEND)
SEND_DATA	**write, writex**	**snawrit**
SEND_ERROR	**ioctl**(SEND_ERROR)	**snactl**(SEND_ERROR)
TEST	Not supported	Not supported

AIX SNA Services/6000 Mapped Conversation Verbs

AIX SNA Services/6000 Mapped Conversation Verbs	
SNA Verb	**Subroutine (Only)**
MC_ALLOCATE	**snalloc**
MC_CONFIRM	**snactl**(CONFIRM)
MC_CONFIRMED	**snactl**(CONFIRMED)
MC_DEALLOCATE	**snadeal**
MC_FLUSH	**snactl**(FLUSH)
MC_GET_ATTRIBUTES	**snactl**(GET_ATTRIBUTE)
MC_POST_ON_RECEIPT	Not supported
MC_PREPARE_TO_RECEIVE	**snactl**(PREPARE_TO_RECEIVE)
MC_RECEIVE_AND_WAIT	**snaread**
MC_RECEIVE_IMMEDIATE	**snaread** (NO_DELAY flag on)
MC_REQUEST_TO_SEND	**snactl**(REQUEST_TO_SEND)
MC_SEND_DATA	**snawrit**
MC_SEND_ERROR	**snactl**(SEND_ERROR)
MC_TEST	Not supported

Option Sets Supported

All SNA products providing an API (application programming interface) must provide API support for a base set of SNA functions and may also provide additional API support for one or more optional sets of functions. In addition to the base-level support, AIX SNA Services/6000 provides support for the following LU 6.2 option sets as defined in the *SNA Transaction Programmer's Reference Manual for LU Type 6.2 (GC30–3084)*:

Flush the LU's send buffer
Allows a program to explicitly cause the LU to flush its send buffer. This option set includes the (MC_)FLUSH verb, but just for local support, since the remote support for this verb is in the LU 6.2 base set of functions.

Get attributes Allows a program to obtain attributes of a mapped conversation. This option set includes the MC_GET_ATTRIBUTES verb; in contrast, the GET_ATTRIBUTES verb for basic conversations is part of the LU 6.2 base set of functions.

Prepare to receive
Allows a program to change the conversation from send state to receive state while simultaneously flushing the LU's send buffer, request confirmation, or request sync point. The transaction program gets control back with the RECEIVE_AND_WAIT verb (or other similar receive verb from an option set). This option set includes the (MC_)PREPARE_TO_RECEIVE verb, but just for local support, since the remote support for this verb is in the LU 6.2 base set of functions.

Receive immediate

Allows a program to receive whatever information is available on a conversation without having to request posting of the conversation. This option set includes the (MC_)RECEIVE_IMMEDIATE verb.

Program reconnect

Allows two cooperating transaction programs to establish a conversation, disconnect the conversation, and then establish the same conversation again to complete the information exchange.

Session-level LU-LU verification

Allows a program or operator to designate the LU-LU passwords (associated with remote LUs) that the local LU uses to verify the identity of a remote LU when a session is activated. This option set relates to the LU_LU_PASSWORD of the DEFINE_REMOTE_LU verb.

Send PIP data

Allows the local program that allocates a conversation to provide PIP (program initialization parameters) to start the remote program. This option set is only for send support. This option set relates to the PIP parameter of the (MC_)ALLOCATE verb and to the PIP parameter of the DEFINE_TP and DISPLAY_TP verbs.

Mapped-conversation LU services component

Allows implementation of a mapped conversation LU services component program, which processes mapped conversation verbs. This option can be used on mapped conversations only and relates to the TYPE parameter of the ALLOCATE, DEALLOCATE, and SEND_ERROR verbs.

Locally known LU names

Allows a program or an operator to specify the locally known names of remote LUs. This option set relates to the LOCALLY_KNOWN_LU_NAME parameter of the DEFINE_REMOTE_LU and DISPLAY_REMOTE_LU verbs. The locally known LU name value may be used in the LU_NAME parameter of the (MC_)ALLOCATE verb and in other verbs having the LU_NAME parameter.

Uninterpreted LU names

Allows a program or an operator to specify the uninterpreted names of remote LUs. This option set relates to the UNINTERPRETED_LU_NAME parameter of the DEFINE_REMOTE_LU and DISPLAY_REMOTE_LU verbs. The uninterpreted LU name value may be used in the LU_NAME parameter of the (MC_)ALLOCATE verb and other verbs having the LU_NAME parameter.

SNA Profiles Used

The following table shows the types of SNA FM (function management) and TS (transmission services) profiles used for the different kinds of sessions that can occur. See *System Networks Architecture Formats* for definitions of these SNA profiles.

LU 6.2 Profiles Used		
Session	**FM Profile**	**TS Profile**
LU-LU	19	7
SSCP-LU	0	1
SSCP-PU	0	1

AIX SNA Services/6000 LUs 1, 2, and 3

LUs 1, 2, and 3 (logical unit types 1, 2, and 3) provide a session between local resources and remote application programs. In this way, SSCP-LU or LU-LU sessions operate on a primary-to-secondary communication basis, with the LUs on the AIX node acting in a secondary role. Local resources (resources connected to this logical unit) may be printers, card readers, card punches, plotters, display devices, or storage devices.

Primary-to-secondary communication allows a remote transaction program to send data to a local device without being involved in the control of the network. LUs 1, 2, and 3 provide a uniform programming interface for buffering data and transmitting the data over the network.

AIX SNA Services/6000 provides both an AIX Operating System subroutine and an SNA library subroutine interface to LUs 1, 2, and 3 functions for application programs. For example, an application program can access the large list of SNA functions by using the following subroutines:

- the **open** subroutine
- the **readx** (or **read**) subroutine
- the **writex** (or **write**) subroutine
- the **close** subroutine
- the **select** subroutine
- the **ioctl** subroutine.

AIX SNA Services/6000 LUs 1, 2, and 3 support conversation protocol boundary (basic conversation verbs), but do not support control operator protocol boundary (control operator verbs) and program protocol boundary (mapped conversation verbs). See the following table for a listing of these verbs and how they are implemented in AIX SNA Services/6000.

Refer to AIX SNA Services/6000 Subroutines in *Communications Programming Concepts* for descriptions of the subroutines listed in the tables. These documents contain complete information about the supported functions.

AIX SNA Services/6000 LU 1, 2, 3 Basic Conversation Verbs

AIX SNA Services/6000 LU 1, 2, 3 Basic Conversation Verbs		
SNA Verb	**Subroutine**	**Subroutine**
ALLOCATE	**ioctl**(ALLOCATE)	**snalloc**
CONFIRM[1]	**ioctl**(CONFIRM)	**snactl**(CONFIRM)
CONFIRMED	**ioctl**(CONFIRMED)	**snactl**(CONFIRMED)
DEALLOCATE	**ioctl**(DEALLOCATE)	**snadeal**
FLUSH	**ioctl**(FLUSH)	**snactl**(FLUSH)
GET_STATUS	**ioctl**(GET_STATUS)	**snactl**(GET_STATUS)
PREPARE_TO_RECEIVE	**ioctl** (PREPARE_TO_RECEIVE)	**snactl** (PREPARE_TO_RECEIVE)
RECEIVE_AND_WAIT	**read, readx**	**snaread**
RECEIVE_IMMEDIATE	**read, readx** (NO_DELAY flag on)	**snaread**(NO_DELAY flag on)
REQUEST_TO_SEND	**ioctl** (REQUEST_TO_SEND)	**snactl**(REQUEST_TO_SEND)
SEND_DATA	**write, writex**	**snawrit**
SEND_ERROR	**ioctl**(SEND_ERROR)	**snactl**(SEND_ERROR)
SEND_FMH[2]	**ioctl**(SEND_FMH)	**snactl**(SEND_FMH)
SEND_STATUS	**ioctl**(SEND_STATUS)	**snactl**(SEND_STATUS)

[1] CONFIRM is used in LU 1, 2, 3 SSCP-LU session, but only in LU 1 LU-LU sessions.

[2] SEND_FMH is used in LU 1 sessions only.

Options Supported

The following options are supported by this implementation of LUs 1, 2, and 3:

Prepare to receive
Allows a program to change the conversation from send state to receive state while simultaneously flushing the LU's send buffer or request confirmation.

Number of columns
Specifies the number of characters across the output device page.

Number of rows
Specifies the number of lines down the output device page.

Notify
A network services command that flows on the SSCP-LU session to provide notification when the power is turned on or off.

Sync level
Used only in LU 1 sessions to allow the user to use the **ioctl**(CONFIRM) or **snactl**(CONFIRM) subroutine.

LUs 1, 2, 3 Implementation Differences

This implementation is different from other implementations of LUs 1, 2, 3 in the following ways:

- This implementation of AIX SNA Services/6000 LUs 1, 2, and 3 uses a verb-level interface rather than a bit-level interface. The table below maps this implementation to LU 1, 2, 3 protocol.
- The AIX node uses the extended version of Exchange ID (XID).
- The Request Maintenance Statistics function is not supported.
- The AIX node requires the format indicator (FI) bit to be set on for all session control (SC) and data flow control (DFC) requests.
- Following an **open** subroutine, the attachment is started. The host sends an ACTPU to the AIX node. After a positive response to the ACTPU is received, the host sends ACTLUs for all active LUs on that physical unit (PU). Thus, connections are started automatically by the host.

LU 1, 2, and 3 Verb to Bit Level Interfacing

AIX Node Usage	LU 1, 2, 3 Protocol
CONFIRM	RQD1
CONFIRMED	Positive response
DEALLOCATE	EB
FLUSH	RQE1
PREPARE_TO_RECEIVE	CD
REQUEST_TO_SEND	SIGNAL
SEND_ERROR	Negative response/Cancel
SEND_STATUS	LUSTAT
RECEIVE (**readx**)	Receive
SEND (**writex**)	Send
ALLOCATE	No equivalent
GET_STATUS	No equivalent

SNA Profiles Used

The following table shows the types of SNA FM (function management) and TS (transmission services) profiles used for the different kinds of sessions that can occur:

LU 1, 2, 3 Profiles Used

Session	FM Profile	TS Profile
LU-LU	3	3
SSCP-LU	0	1
SSCP-PU	0	1

AIX SNA Services/6000 LU 0

The LU 0 Subsystem provides an application program interface (API) for LU 0 Primary support and for LU 0 Secondary support. LU 0 logical units provide session support for host (AIX LU 0 acting as Secondary) and subordinate (AIX LU 0 acting as Primary) communications. The API can be accessed via a subroutine interface.

These subroutines allow you to open, send, receive, close, and control the session from within your application. The Secondary and Primary support have separate APIs for use by the application program:

- Secondary
  - the **lu0opens** subroutine
  - the **lu0closes** subroutine
  - the **lu0reads** subroutine
  - the **lu0writes** subroutine
  - the **lu0ctls** subroutine.
- Primary
  - the **lu0openp** subroutine
  - the **lu0closep** subroutine
  - the **lu0readp** subroutine
  - the **lu0writep** subroutine
  - the **lu0ctlp** subroutine.

SNA Profiles Used

The following table shows the types of SNA FM (function management) and TS (transmission services) profiles used for the different kinds of sessions that can occur. See *Systems Network Architecture Formats* for definitions of these SNA profiles.

LU 0 Profiles Used		
Session	**FM Profile**	**TS Profile**
LU-LU	3	3
LU-LU	4	4
SSCP-LU	0	1
SSCP-PU	0	1

AIX SNA Services/6000 Generic SNA Device Driver

The generic SNA device driver provides a special interface to AIX SNA Services/6000. This special interface allows the generic SNA application to utilize the PU Services function (PU 2.1 only) of AIX SNA Services/6000.

The generic SNA application can use this special interface to establish an AIX SNA Services/6000 attachment or share the same AIX SNA Services/6000 attachment with AIX SNA Services/6000. For example, a generic SNA LU_0 secondary application can run simultaneously with a 3270 emulation application to the Host system over the same AIX SNA attachment. Refer to AIX SNA Services/6000 Subroutines in *Communications*

Programming Concepts for descriptions of other programming interfaces for the following subroutines.

The generic SNA device driver provides the following API subroutines to interface with AIX SNA Services/6000:

open Specifies an AIX SNA Services/6000 attachment profile name in the open path to open a file descriptor for an AIX SNA Services/6000 attachment.

close Closes a file descriptor.

read Receives data from a file descriptor.

write Sends data to a file descriptor.

select Waits for file descriptors until data is available or until an exception condition occurs for the file descriptors.

ioctl Performs the following command options:

HIER_RESET_RSP Responds to a hierarchical reset from the PU Services of AIX SNA Services/6000.

INOP_RSP Responds to an INOP from the PU Services of AIX SNA Services/6000.

IOCINFO Requests the type of device.

Customizing AIX SNA Services/6000

Once you have installed AIX SNA Services/6000 and the supporting devices and data links onto the system and then properly configured the devices, AIX SNA Services/6000 is ready to run. However, AIX SNA Services/6000 cannot operate correctly on the network unless you tell it about your network and which programs to run, using the network. In Customizing AIX SNA Services/6000 you will learn:

- How to backup SNA profiles
- How to describe your network to AIX SNA Services/6000 in detail.

To use the information in this section, you must install AIX SNA Services/6000, according to the instructions in How to Install System Network Architecture Services/6000 Licensed Program on page 13–45. In addition, you should understand the background information presented in Introducing AIX SNA Services/6000 on page 13–1, with particular emphasis on the sections How AIX SNA Services/6000 Stores Information on page 13–20 and AIX SNA Services/6000 Network Names and Addresses on page 13–24.

How to Install System Network Architecture Services/6000 Licensed Program

You can install the SNA Services/6000 licensed program the following ways:

- From a preloaded disk
- From a tape
- From a network server
- From a diskette.

The following sections describe how the licensed program is packaged, the system requirements for installation, and procedures for installing the licensed program.

How the AIX SNA Services/6000 Licensed Program is Packaged

The SNA Services/6000 licensed program consists of the following parts:

sna.sna.obj	Contains the SNA Services/6000 Base program.
sna.lu0.obj	Contains the SNA Services/6000 LU0 facility.
snam*m_Language*.msg	Contains the messages and helps in the specified *Language* for the run time environment. If you install multiple languages for a product, be sure that you install the preferred (or primary) language first.

The licensed program may also contain update files.

Prerequisite Tasks or Conditions

- You must log in as root user.
- You must have installed the AIX Base Operating System (BOS) Runtime (part of the IBM AIX Base Operating System licensed program).
- You must have installed the Base Application Development Toolkit (part of the AIX Base Operating System licensed program) before you attempt to statically bind any SNA Services/6000 modules.

- You must install the **sna.sna.obj** (the SNA Services/6000 Base program) prior to the **sna.lu0.obj**.
- The SNA system must not be active. To determine if the system is active, enter:

  ```
  ls src –s sna
  ```

  If the subsystem is active, enter the following to terminate it:

  ```
  stop src –s sna
  ```

Procedure to Install for Preloaded Disk

1. Enter the following on an AIX command line:

   ```
   smit instupdt
   ```

   This command invokes the System Management Interface Tool (SMIT), which presents a menu driven environment for the installation process.
2. Follow the directions and answer the prompts in the SMIT Install menus. Select **sna.sna.obj**, **sna.lu0.obj**, **snam***m_Language***.msg**, and any updates for the licensed program.

   Messages display as each part successfully completes installing.

Procedure to Install from Tape

1. Insert the tape containing **sna.sna.obj**, or **sna.lu0.obj** into the tape drive.
2. Enter the following on an AIX command line:

   ```
   smit instupdt
   ```

   This command invokes the System Management Interface Tool (SMIT), which presents a menu driven environment for the installation process.
3. Follow the directions and answer the prompts in the SMIT Install menus. Select **sna.sna.obj**, **sna.lu0.obj**, **snam***m_Language***.msg**, and any updates for the licensed program.

   Messages display as each part successfully completes installing.

Procedure to Install over a Network

1. Refer to How to Install over a Network in the *Installation Kit for IBM AIX Version 3 for RISC System/6000* for instructions on how to set up the server and how to down load files to the client.
2. Select Optional Program Products from the System Startup menu.
3. Follow the directions and answer the prompts in the SMIT Install menus. Select the files that contain **sna.sna.obj**, **sna.lu0.obj**, **snam***m_Language***.msg**, and any updates for the

   Messages display as each part successfully completes installing.

Procedure to Install from Diskette

1. Insert the first **sna.sna.obj** or **sna.lu0.obj**, diskette into the diskette drive.
2. Enter the following on an AIX command line:

   ```
   smit instupdt
   ```

   This command invokes the System Management Interface Tool (SMIT), which presents a menu driven environment for the installation process.

3. Follow the directions and answer the prompts in the SMIT Install menus. First install **sna.sna.obj**, or **sna.lu0.obj**, then install **snam***m_Language***.msg**, and finally apply the updates.

   Messages display as each part successfully completes installing.

Related Information

The **installp** command, **updatep** command, **instupdt** command, and the **smit** command. For information about these commands, use the InfoExplorer information retrieval facility. This information is also included in the *AIX Commands Reference for IBM RISC System/6000.*

Installation Instructions for IBM AIX Version 3 for RISC System/6000.

The System Manager Interface Tool (SMIT) Overview in *General Concepts and Procedures.*

Installing AIX SNA Services/6000: Additional Information

To install AIX SNA Services/6000 no devices, applications, or files need be inactive, nor do any subsystems need to be stopped. The system can be in either single user or multiuser mode and does not need to be in maintenance mode.

You do not need to reconfigure the kernel for the install to take effect (this automatically happens). You do not need to initially configure the AIX SNA Services/6000 profiles because the **peu** command is automatically invoked during installation. However, you will surely want to customize your profiles.

Refer to Customizing AIX SNA Services/6000 on page 13–45 for more information on configuring AIX SNA Services/6000 profiles. Refer to Creating Default Profiles with the PEU Utility on page 13–48 for more information on creating an alternate set of AIX SNA Services/6000 profiles.

You do not need to re-IPL for AIX SNA Services/6000 to become useable, however if AIX SNA Services/6000 is running, you do need to stop it to do the installation and then restart it for the install to take effect. Refer to Starting and Stopping AIX SNA Services/6000 on page 13–220 for more information on starting and stopping AIX SNA Services/6000.

Refer to Defining a Network to AIX SNA Services/6000 (System Management Interface Tool (SMIT) Method) on page 13–50, and AIX SNA Services/6000 Subroutines for Transaction Program Conversations in *Communications Programming Concepts*for more information on related topics.

Backing Up SNA Configuration Profiles

You should periodically make a backup copy of the SNA profiles to protect against loss of the information. In addition to the normal periodic backup procedure, backup these profiles under any of the following conditions:

- After you create a new set of profiles
- Before and after you update or change a current set of profiles
- Before and after you change the security level of AIX SNA Services/6000
- Before and after you change the SNA Communications Authority Password (to protect against forgetting the password).

You must use a special procedure to backup the profiles. Enter the following commands on the AIX Operating System command line to backup the SNA profiles in the standard directory:

```
cd /usr/lpp/sna/objrepos
ls –a | backup –i
```

To backup profiles in a different directory, use the **cd** command to change to that directory, and then use the **backup** command. This procedure dumps all of the profiles in the selected directory to the diskette, **/dev/fd0**. If you use another method of backing up the files, make sure that the method you choose backs up the hidden files (those files that begin with a . (period)) as well as the normal files.

If you do not backup the SNA profiles and the database becomes corrupted, you can create a new set of default profiles with the **peu** utility. However, to do this, you must delete or move all the files in the **/usr/lpp/sna/objrepos** directory where the profiles reside. Refer to Creating Default Profiles with the PEU Utility on page 13–48 for more information.

Restoring a Backup Copy of SNA Profiles

To restore SNA profiles that were backed up to diskette, using the procedure described in Backing Up SNA Configuration Profiles on page 13–47, use the following commands:

```
cd /usr/lpp/sna/objrepos
restore –x
```

If a directory other than **/usr/lpp/sna/objrepos** was backed up, use the **cd** command to change to that directory instead.

When you issue these commands, SNA must not be active.

Creating Default Profiles with the PEU Utility

The **peu** utility runs as part of the **snaserv** installation procedure to create the initial default profiles. Ordinarily, you should not need to run the **peu** program again. If, however, your profiles database becomes corrupted or if you want to configure an alternate set of profiles, you can create a new set of default profiles with the **peu** utility.

Note: The utility configures the default directory, **/usr/lpp/sna/objrepos**. Delete or move all files, including hidden files, from this directory before you run the utility. If you are creating an alternate set of profiles, be sure to move the existing profiles to a new directory, as the whole directory is overwritten.

The **peu** program can create default profiles for any of the profile types necessary for running SNA.

To use the utility, issue the following commands:

```
cd /usr/lpp/sna/bin
peu
```

If storage space is a problem, you can delete the **peu** program after the initial installation is complete and restore it from the install diskettes at the time you create the new set of default profiles.

Verifying Profiles

The **verifysna** command, available through the System Management Interface Tool (SMIT) or the command line, checks your profile database for inconsistencies across different profiles. It checks that all profiles referenced exist on the database and that values do not conflict.

You should verify your profiles after you finish modifying them. You can also use verification while you are creating a new profile database or making extensive changes to existing profiles. If you change only a connection profile, you should still use the **verifysna** command to make sure that all references to other profiles exist.

Verification can take several minutes. The **verifysna** command continues checking for inconsistencies after finding the first error. For each error message received, note the necessary information, make the necessary changes, and then run the **verifysna** command again to ensure that problems are corrected.

Using the **verifysna** command, you can check a list of profiles to determine if they are correct. The **verifysna** command does not guarantee that all network parameters are accurate. Only those parameters that can be checked within the local system are verified.

The following procedure describes how to verify SNA configuration profiles, using the SMIT Interface.

Verifying Profile Lists

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Verify SNA Configuration Profiles.`

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

The generated command for this dialog is:

```
verifysna
```

External Device Configuration Dependencies

AIX SNA Services/6000 has a dependency on the correct setting of the `Receive Data Transfer Offset (RDTO)` field in the communications adapter device definitions. The RDTO field defines the offset at which the adapter is to start writing received data into a receive buffer. RDTO *must* be set to 92 (decimal) to ensure the proper operation of AIX SNA Services/6000. For information on how to set the RDTO field, refer to the appropriate description of defining an Ethernet Adapter, Token-Ring Adapter, Multiprotocol Adapter, or X.25 Adapter in the following books:

RISC System/6000 Power Station and Power Server Hardware Technical Reference Options and Devices (SA23–2646).

The 4-Port Multiprotocol Interface Adapter Technical Reference (S33F–5337).

X.25 Co-Processor/2 Technical Reference (S16F–1879).

Defining a Network to AIX SNA Services/6000 (System Management Interface Tool (SMIT) Method)

To define your network to AIX SNA Services/6000, you must provide information to the system that describes the characteristics of the network and the local and remote software that communicates over the network. To provide this information, you must fill in forms or *profiles* that are displayed by the system. Profiles provide a convenient means for entering all the data needed to define the network to AIX SNA Services/6000 so that the local system can begin communicating on the network as soon as possible.

The following paragraphs describe each of the profiles that you must fill in and the information required by each profile. AIX SNA Services/6000 Customization Forms on page 13–257 provides forms for each profile so that you can fill in the information about your network before actually entering the information into the system. Using the forms allows you to prepare ahead of time for customizing AIX SNA Services/6000 and also provides a written record of the customizing information.

To decide which profiles to fill in, you need to know the following characteristics about the installation:

- Which SNA LU (logical unit) interface does your system use?
  - LU 0
  - LU 1
  - LU 2
  - LU 3
  - LU 6.2.
- To what physical interface is the system connected?
  - Standard Ethernet LAN
  - Token-Ring LAN
  - IEEE 802.3 Ethernet LAN
  - X.25/QLLC Packet Network
  - Synchronous EIA232D PTN
  - Synchronous Smart Modem PTN
  - X.21 Data Network
  - V.25 bis
  - V.35
  - EIA422A.

Use the tables on the following pages to determine which profiles are required for your installation.

For:	**Use Profiles Listed In:**
LU 0	LU 0 Installations on page 13–52
LU 1, LU 2, LU 3, LU 6.2	LU 1, 2, 3, and 6.2 Installations on page 13–52
LU 6.2 (only)	LU 6.2 Installations on page 13–52
Ethernet installations	Ethernet Installations on page 13–53
Token-Ring installations	Token-Ring Installations on page 13–53
IEEE 802.3 Ethernet installations	IEEE 802.3 Ethernet Installations on page 13–53
X.25 installations	X.25 Installations on page 13–54
Smart Modem installations	Smart Modem Installations on page 13–54
EIA232D installations	EIA232D Installations on page 13–54
X.21 installations	X.21 Installations on page 13–54
EIA422A	EIA422A Installations on page 13–55
V.35	V.35 Installations on page 13–55
V.25 bis	V.25 bis Installations on page 13–55.

After determining which profiles are required, use the text descriptions indicated in the tables to help determine the proper values to enter into the indicated forms. Then use the procedures described in the text descriptions to enter the information from the forms into profiles on the system.

Notes:

1. You must have root user authority (UID = 0) to add, change, or delete profiles whose last seven characters are `DEFAULT`. These profiles contain the system default values and are protected from accidental changes.
2. When choosing names for profiles, be aware that the names of individual profiles within each type of profile cannot be the same. For example, all connection profiles must have different names. In addition, the name of any profile within the following groups of profile types cannot be the same as the name of another profile in that same group:
   - TPN and RTPN profiles
   - TPN list and RTPN list profiles
   - Logical link profiles (SDLC, Ethernet, Token-Ring, 802.3 Ethernet, X.25)
   - Physical link profiles (Ethernet, Token-Ring, 802.3 Ethernet, X.25, EIA232D, Smart Modem, X.21, EIA422A, V.23 bis, and V.35).

LU 0 Installations

Profiles Required for LU 0 Installations		
Profile	**Text Description**	**Form To Fill In**
SNA	Defining AIX SNA Services/6000 Characteristics	SNA Profile
Connection	Defining Connection Characteristics	Connection Profile
Local Logical Unit	Defining Generic LU Local Logical Unit Characteristics	Local Logical Unit Profile
Attachment	Defining Attachment Characteristics	Attachment Profile
Control Point	Defining Physical Unit Characteristics	Control Point Profile

LU 1, 2, 3, and 6.2 Installations

Profiles Required for LU 1, 2, 3, and 6.2 Installations		
Profile	**Text Description**	**Form To Fill In**
SNA	Defining AIX SNA Services/6000 Characteristics	SNA Profile
Connection	Defining LU Type x (1,2,3,6.2) Connection Characteristics	Connection Profile
Local Logical Unit	Defining LU Type x (1,2,3,6.2) Local Logical Unit Characteristics	Local Logical Unit Profile
Attachment	Defining Attachment Characteristics	Attachment Profile
Control Point	Defining Physical Unit Characteristics	Control Point Profile

LU 6.2 Installations

Additional Profiles Required for LU 6.2 Installations		
Profile	**Text Description**	**Form To Fill In**
Mode	Defining LU Type 6.2 Session Characteristics	Mode Profile
Mode List	Defining LU Type 6.2 Modes for a Session	Mode List Profile
Transaction Program Name	Defining LU Type 6.2 Application Program Characteristics	Transaction Program Name Profile

Transaction Program Name List	Defining LU Type 6.2 Transaction Programs for a Session	TPN List Profile
Remote Transaction Program Name	Defining LU Type 6.2 Remote Application Program Characteristics	Remote Transaction Program Name Profile
Remote Transaction Program Name List	Defining LU Type 6.2 Remote Transaction Programs for a Session	RTPN List Profile

Standard Ethernet Installations

Profiles Required for Standard Ethernet Installations		
Profile	**Text Description**	**Form To Fill In**
Standard Ethernet Logical Link	Defining Standard Ethernet Logical Link Characteristics	Standard Ethernet Logical Link Profile
Standard Ethernet Physical Link	Defining Standard Ethernet Physical Link Characteristics	Standard Ethernet Physical Link Profile

Token-Ring Installations

Profiles Required for Token-Ring Installations		
Profile	**Text Description**	**Form To Fill In**
Token-Ring Logical Link	Defining Token-Ring Logical Link Characteristics	Token-Ring Logical Link Profile
Token-Ring Physical Link	Defining Token-Ring Physical Link Characteristics	Token-Ring Physical Link Profile

IEEE 802.3 Ethernet Installations

Profiles Required for IEEE 802.3 Ethernet Installations		
Profile	**Text Description**	**Form To Fill In**
IEEE 802.3 Ethernet Logical Link	Defining IEEE 802.3 Ethernet Logical Link Characteristics	IEEE 802.3 Logical Link Profile
IEEE 802.3 Ethernet Physical Link	Defining IEEE 802.3 Ethernet Physical Link Characteristics	IEEE 802.3 Physical Link Profile

X.25 Installations

Profiles Required for X.25 Installations		
Profile	**Text Description**	**Form To Fill In**
QLLC Logical Link	Defining QLLC Logical Link Characteristics	QLLC Logical Link Profile
X.25 Physical Link	Defining X.25 Physical Link Characteristics	X.25 Physical Link Profile

EIA232D Installations

Profiles Required for EIA232D Installations		
Profile	**Text Description**	**Form To Fill In**
SDLC Logical Link	Defining SDLC Logical Link Characteristics	SDLC Logical Link Profile
EIA232D Physical Link	Defining EIA232D Physical Link Characteristics	EIA232D Physical Link Profile

Smart Modem Installations

Profiles Required for Smart Modem Installations		
Profile	**Text Description**	**Form To Fill In**
SDLC Logical Link	Defining SDLC Logical Link Characteristics	SDLC Logical Link Profile
Smart Modem Physical Link	Defining Smart Modem Physical Link Characteristics	Smart Modem Physical Link Profile

X.21 Installations

Profiles Required for X.21 Installations		
Profile	**Text Description**	**Form To Fill In**
SDLC Logical Link	Defining SDLC Logical Link Characteristics	SDLC Logical Link Profile
X.21 Physical Link	Defining X.21 Physical Link Characteristics	X.21 Physical Link Profile

EIA422A Installations

Profiles Required for EIA422A Installations

Profile	Text Description	Form To Fill In
SDLC Logical Link	Defining SDLC Logical Link Characteristics	SDLC Logical Link Profile
EIA422A Physical Link	Defining EIA422A Physical Link Characteristics	EIA422A Physical Link Profile

V.35 Installations

Profiles Required for V.35 Installations

Profile	Text Description	Form To Fill In
SDLC Logical Link	Defining SDLC Logical Link Characteristics	SDLC Logical Link Profile
V.35 Physical Link	Defining V.35 Physical Link Characteristics	V.35 Physical Link Profile

V.25 bis Installations

Profiles Required for V.25 bis Installations

Profile	Text Description	Form To Fill In
SDLC Logical Link	Defining SDLC Logical Link Characteristics	SDLC Logical Link Profile
V.25 bis Physical Link	Defining V.25 bis Physical Link Characteristics	V.25 bis Physical Link Profile

Defining SDLC Attachment Characteristics

The attachment profile contains fields that both associate other defined profiles with the attachment of the LU to the network and define the type of network being used. AIX SNA Services/6000 uses the information in this profile to open an attachment to the described network through a hardware adapter. Each attachment to a network must have an attachment profile defined for it.

Select this value if the network uses Synchronous Data Link Control (SDLC) as its data link protocol. Also, select this value if you are setting up a new network that does not use one of the other specified link types.

The following procedure describes how to add an SDLC attachment profile, using the SMIT Interface

Entering SDLC Attachment Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `SDLC.`
7. From the next SMIT menu, select `SDLC Attachment.`
8. From the next SMIT menu, select `Add a Profile.`
9. This displays the `Add SNA SDLC Attachment Profile` dialog. Add any names and change any default values necessary to ensure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, `Change an Alias for a Profile`, and `Change Generic LU Address Registration.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t attachment –w PhysicalLinkType ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the attachment profile. Once the new attachment profile appears on the screen, you must enter some additional information to correlate the attachment with the other profiles that describe it. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press Enter (`Enter=Do`) to save the new profile with the modified parameters. The following paragraphs supply information to help choose the values that best describe the application program.

Add SNA SDLC Attachment Profile Dialog

PROFILE name	....
CONTROL POINT profile name	CDEFAULT
LOGICAL LINK profile name	SDEFAULT
PHYSICAL LINK profile name	RDEFAULT
STOP ATTACHMENT on inactivity?	no
If yes, inactivity TIMEOUT (0 – 10 Min)	0
LU address REGISTRATION?	no
If yes, LU address REGISTRATION PROFILE name	....
STATION type	secondary
If primary,	
REMOTE SECONDARY station address (1 – 255)	1
PHYSICAL LINK type	EIA232D
If Smart Modem or V.25 bis,	
Modem COMMAND SEQUENCE	....
If X.21, NETWORK type	switched
If switched, CALL type	listen
If call, SELECTION sequence	....

Figure 1. Add SNA SDLC Attachment Profile Dialog

PROFILE name

This field requests a name for the new attachment profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

CONTROL POINT profile name

This field provides the name of the control point profile that defines the node ID of the physical unit associated with this attachment. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field. You must create a control point profile that has the name that you provide in this field. Refer to Defining Physical Unit Characteristics on page 13–215 for information about creating this profile.

LOGICAL LINK profile name

This field provides the name of the logical link profile that defines the characteristics of the data link protocol that implements the network. You must create a logical link profile that has the name that you provide in this field.Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

Refer to Defining SDLC Logical Link Characteristics on page 13–62 for information about creating the logical link profile.

PHYSICAL LINK profile name

This field provides the name of the physical link profile that defines the characteristics of the physical port for the network. You must create a physical link profile for the name that you provide in this field. Refer to one of the following topics for information on creating the physical link profile:

- Defining EIA232D Physical Link Characteristics on page 13–81
- Defining Smart Modem Physical Link Characteristics on page 13–86
- Defining X.21 Physical Link Characteristics on page 13–91
- Defining EIA422A Physical Link Characteristics on page 13–96
- Defining V.25 bis Physical Link Characteristics on page 13–100
- Defining V.35 Physical Link Characteristics on page 13–105

Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on choosing a name for this field.

STOP ATTACHMENT on inactivity?

This field specifies whether AIX SNA Services/6000 should stop the attachment if no connections are active on the attachment for a specified period of time. This choice affects system performance because each open attachment uses system resources. If the attachment is idle, the resources may be used more efficiently elsewhere.

However, restarting the attachment takes time to resolve the profiles associated with a connection, and you may want to keep the attachment active to avoid the delay. In addition, if the link is not terminated when the stop occurs, cleanup procedures that require interaction with other nodes may not be completed. If this happens, put the link back into an active state for a normal stop to occur.

In general, unless keeping the attachment open is very important, select `yes` and choose a value for the time-out period.

`yes` Select this value to stop the attachment after a period of inactivity. If you select this value, you must specify the length of the inactivity period in the next field.

`no` Select this value to remain attached to the remote node regardless of how long the connection remains idle.

Inactivity TIMEOUT

This field should only be changed and is required when the `STOP ATTACHMENT on inactivity?` field is `yes`. Enter a value in the specified range (0 to10) for the number of minutes to wait before stopping the attachment. The actual waiting period is within 30 seconds of the specified number of minutes.

LU address REGISTRATION?

This field specifies whether LU addresses is registered for use by the attachment with the generic SNA application.

`yes` Select this value if you are using generic SNA and LU addresses are registered. Enter the name of the profile containing the list of registered LU addresses in the `LU address REGISTRATION PROFILE name` field.

`no` Select this value if you are not using generic SNA and do not need registered LU addresses.

LU address REGISTRATION PROFILE name

This field should only be changed if the `LU address REGISTRATION?` field is `yes`. This field provides the name of the LU address registration profile name that contains a list of LU addresses to be registered for use by this attachment with generic SNA.

STATION type

This field allows you to define the role of the local station with respect to other stations on the network. Select the role that fits the operation of the local station within the network. The value you select must match the `STATION type` field that you select in the SDLC logical link profile.

`primary` Select this option if the local station operates as an SDLC *primary* station. A primary station has responsibility for the data link and issues commands to secondary stations as well. For example, a control station in a multipoint network is a primary station. If you select this option, specify the remote secondary station address in the next field.

`secondary` Select this option if the local station operates as an SDLC *secondary* station. A secondary station is one that responds to requests from another station (the primary station) and has little control over data link operations.

`negotiable` Select this option if the local station can operate as either a primary or a secondary station, depending upon the specific remote station to which it is connected. That is, the station type is negotiated.

REMOTE SECONDARY station address

This field should only be changed when the `STATION type` field is `primary`. This address specifies the station address for the remote secondary station connected to this link. This is the address field in the SDLC data stream that identifies the secondary station. Enter a decimal integer, within the specified range, that matches the station address assigned to the secondary station. The value entered in this field must match the value entered in the `Local SECONDARY STATION address` field in the SDLC logical link profile for the remote station. See Defining SDLC Secondary Logical Link Characteristics on page 13–63 for a description of that field.

PHYSICAL LINK type

This field designates the type of physical link being used. Use the following information to determine which selections to make:

`EIA232D` Select this type if the connection to the network uses the interface defined by Electronic Industries Association (EIA) standard EIA232D. This interface is used for directly connecting systems over short distances (less than 60 meters) or for connecting to a public switched telephone network for long distances, using a synchronous modem such as the IBM 386X synchronous modem.

`SMART MODEM` Select this type if the connection to the public switched telephone network uses a synchronous modem that employs the Smart Modem control protocol. If you select this type, specify the modem command sequence as the next field.

`X.21` Select this type if the connection to the public data network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation X.21. This interface is used to connect to the national data networks in many European countries and Japan.

`EIA422A` Select this type if the connection to the network uses the interface defined by Electronic Industries Association (EIA) standard EIA422A. This interface is used for directly connecting systems over medium distances (less than 1.2 kilometers) or for connecting to a public switched telephone network for long distances, using a synchronous modem such as the IBM 386X synchronous modem.

`V.35` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.35. This interface is used to connect to the national telephone networks in many European countries and Japan.

`V.25 bis` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.25 bis. This interface is used to connect to the national telephone networks in many European countries and Japan.

Modem COMMAND SEQUENCE

This field should only be changed when the PHYSICAL LINK type field is SMART MODEM. The modem command sequence is a series of up to 79 ASCII characters that control the modem's call-establishment and data-transfer options. This sequence is transferred to the modem as a single character string, using asynchronous communications when the attachment is first started. A terminating carriage return is automatically added to the end of the sequence. Once the modem's data set ready (DSR) circuit becomes active, the modem is switched to a synchronous mode for the remainder of the call.

For example, to call a remote station, using a telephone number of 555–1234, the following command sequence might be used:

```
AT &S1 &M1 S25=0 DT 5551234
```

For an incoming call, the following command sequence might be used:

```
AT &S1 &M1 S25=0 S14=10 S0=2
```

NETWORK type

This field should only be changed if the PHYSICAL LINK type field is X.21, EIA422A, or V.35. This field indicates the type of transmission lines that the network uses.

switched — Select this parameter if the X.21, EIA422A, or V.35 port is connected to a public data network switched-circuit service CCITT X-Series interface (also called Telegraph and Telephone Facility-L).

nonswitched — Select this parameter if the X.21, EIA422A, or V.35 port is connected to a public data network nonswitched-circuit service CCITT X-Series interface (also called Telegraph and Telephone Facility-N).

CALL type

This field should only be changed if the NETWORK type field is switched. The CALL type field indicates whether the local station initiates a connection or receives requests for a connection from another station. Select one of the following values that describes how the local station operates:

call — Select this value to indicate that the local station initiates a connection by calling another station. If you choose this field, specify the selection sequence in the next field.

listen — Select this value to indicate that the local station does not initiate a connection, but waits for a remote station to make a connection with it.

SELECTION sequence

This field should only be changed when the CALL type field is call. The selection sequence is a series of up to 255 ASCII characters that determine the destination of the information on the network. Only the digits 0 to 9 and the special characters * (asterisk), + (plus), , (comma), – (hyphen), . (period), and / (slash) are allowed. A + (plus) is added to the end of the sequence as a terminator.

Defining SDLC Logical Link Characteristics

The SDLC logical link profile defines characteristics of the SDLC (Synchronous Data Link Control) line protocol. The system uses the information in this profile to determine how the line protocol operates on the associated attachment. If the network uses the SDLC protocol, you must define at least one SDLC logical link profile.

You can define more than one SDLC logical link profile, but only one set of conditions or profile can be active on a particular attachment. The attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) designates the logical link profile that is active on a particular attachment.

The following procedure describes how to add an SDLC logical link profile, using the SMIT Interface

Entering SDLC Logical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.

2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`

3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `SDLC.`
7. From the next SMIT menu, select `SDLC Data Link Control.`
8. From the next SMIT menu, select `SDLC Logical Link.`
9. From the next SMIT menu, select `SDLC Secondary Logical Link, SDLC Primary Logical Link,` or `SDLC Negotiable Logical Link.`
10. From the next SMIT menu, select `Add a Profile.`
11. This displays the `Add SNA SDLC Secondary Logical Link Profile, Add SNA SDLC Primary Logical Link Profile,` or `Add SNA SDLC Negotiable Logical Link Profile` dialogs. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base commands for these dialogs are:

```
mksnaobj –t log_sdlc –b secondary –w PhysicalLink ProfileName

mksnaobj –t log_sdlc –b primary –w PhysicalLink ProfileName

mksnaobj –t log_sdlc –b negotiable –w PhysicalLink ProfileName
```

Changing the Default Characteristics

The next three profile examples show the information fields that comprise the SDLC logical link profiles. The first profile example on page 13–64 shows the fields for the SDLC Secondary Logical Link Profile, the second profile example on page 13–68 shows the fields for the SDLC Primary Logical Link Profile, and the third profile example on page 13–75 shows the fields for the SDLC Negotiable Logical Link Profile. Once the new SDLC logical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters. The following paragraphs supply information to help choose the values that best describe your network.

Defining SDLC Secondary Logical Link Characteristics

Select this option if the local station operates as an SDLC secondary station. A secondary station is one that responds to requests from another station (the primary station) and has little control over data link operations.

Use the following information to help determine the correct information for each field.

Changing the Default Characteristics

The SDLC Secondary Logical Link profile example shows the information fields that comprise the SDLC secondary logical link profile. Once the new SDLC logical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA SDLC Secondary Logical Link Profile Dialog

PROFILE name	....
PHYSICAL LINK type	EIA232D
TRANSMIT window count	7
RETRANSMIT count (1–50)	10
Retransmit THRESHOLD (0–100)	10
DROP LINK on inactivity?	no
FORCE DISCONNECT timeout (1–600 seconds)	120
DEFINITION of maximum I-FIELD size	system_defined
If user-defined, max. I-FIELD SIZE (265–30729)	265
TRACE Link?	no
If yes, TRACE SIZE	short
Secondary INACTIVITY timeout (1–120 seconds)	30
Local SECONDARY STATION address	1

Figure 2. Add SNA SDLC Secondary Logical Link Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

PROFILE name

This field requests a name for the new logical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) for the attachments that use this SDLC logical link profile. Do not change this name without changing it throughout the entire profile.

PHYSICAL LINK type

This field designates the type of physical link being used. Use the following information to determine which selections to make.

`EIA232D` Select this type if the connection to the network uses the interface defined by Electronic Industries Association (EIA) standard EIA232D. This interface is used for directly connecting systems over short distances (less than 60 meters) or for connecting to a public switched telephone network for long distances, using a synchronous modem such as the IBM 386X synchronous modem.

`SMART MODEM` Select this type if the connection to the public switched telephone network uses a synchronous modem that employs the Smart Modem control protocol. If you select this type, specify the modem command sequence as the next field.

`X.21` Select this type if the connection to the public data network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation X.21. This interface is used to connect to the national data networks in many European countries and Japan.

`EIA422A` Select this type if the connection to the network uses the interface defined by Electronic Industries Association (EIA) standard EIA422A. This interface is used for directly connecting systems over medium distances (less than 1.2 kilometers) or for connecting to a public switched telephone network for long distances, using a synchronous modem such as the IBM 386X synchronous modem.

`V.35` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.35. This interface is used to connect to the national telephone networks in many European countries and Japan.

`V.25 bis` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.25 bis. This interface is used to connect to the national telephone networks in many European countries and Japan.

TRANSMIT window count

This field specifies the maximum number of outstanding SDLC information frames that may be sent before a response from the remote station is required. The response from the remote station includes the number of frames that it received to ensure that all frames that were sent arrived at the intended destination.

The value that you supply for this field affects throughput rate on the data link. The higher the number entered for this field, the better the throughput rate. The effect on throughput rate is due to the time required to reverse the direction of data flow. Fewer changes in the direction of data flow result in more time available to transmit data. However, if the data link frequently loses frames, select a lower value for the `TRANSMIT window count` field to save the time required to retransmit larger packages of data when an error occurs.

If you do not have a value established, use the default value. Adjust the value later to allow for data link conditions.

RETRANSMIT count

This field specifies the number of contiguous information frame bursts containing the same data that the local station retransmits before it declares a permanent transmission error. This nonproductive sequence can occur if the remote station can respond to the information frame bursts but does not accept any of the data. For example, if this field contains a value of 10 and the remote station indicates both that:

- It accepted none of the data, and
- It can still receive data

then the local station transmits the data again up to 9 more times. If the remote station still does not accept the data, the local station starts inactivity procedures (see SDLC Inactivity Procedures on page 13–80).

The value entered in this field affects transmission throughput and the ability to maintain a connection with a remote station. If you specify a high value, only the worst link conditions result in disconnecting the link. However, the transmission may take a long time due to the large number of attempts to transmit the data if errors do occur. If you specify a very low value, minor transmission difficulties may result in disconnecting the link. If you do not have a value specified for the network, start with the default value and adjust it later to allow for network conditions.

Retransmit THRESHOLD

This field specifies the number of information frame retransmissions allowed as a percentage of total information frame transmissions (sampled only after a block of information frames has been sent). The specified percentage equals the maximum rate of retransmissions allowed above which the system declares that a temporary transmission error has occurred. When a temporary transmission error occurs, the system logs an entry in the error log.

Use this field to provide feedback about the performance of a particular link. If the field is set too low, the system error log contains information that may not indicate a link problem. If the field is set too high, the error log may not contain any link information. If you do not have a value specified for the network, start with the default value and adjust it later to provide a meaningful indication of link performance in the error log.

DROP LINK on inactivity?

This field specifies whether AIX SNA Services/6000 should disconnect the remote station if the local station does not receive a transmission on the link for a specified period of time. This field affects system performance because each open link uses system resources.

If the link is idle, then the resources may be used more efficiently elsewhere. However, restarting the link takes time to resolve the profiles associated with a connection, and you may want to keep the link active to avoid the delay. Unless keeping the link open is very important, select `yes` and choose a value for the `Secondary INACTIVITY timeout` field.

`yes` Select this value to drop the link after a period of inactivity.

`no` Select this value to receive notification of inactivity and then remain active regardless of how long the link remains idle.

Refer to SDLC Inactivity Procedures on page 13–80 for more SDLC inactivity information.

FORCE DISCONNECT timeout

This field specifies the number of seconds that the system should wait after requesting a disconnect from the link (DISC), before the system forces the disconnect. The value that you choose varies with the network. Choose a value that represents a reasonable amount of time for the network, within its abilities, to respond to a disconnect without tying up local system resources for an extended period while waiting for the disconnect.

This value can range from 1 to 600 seconds. If you do not know what value to use, select the default value, 120 seconds, adjusting it as necessary to respond to actual network performance.

DEFINITION of maximum I-FIELD size

This field specifies how the maximum information field size within a link packet is determined, using the following options.

`system_defined`
: This is the default selection that allows the maximum I-field size to be determined by the system by way of XID exchanges with the remote partner. The maximum value negotiated is 30,729 bytes, which may be reduced by factors such as the buffer size of the port and the buffer capability of the remote station.

`user_defined`
: This option allows you to provide your own value for the maximum I-field size.

Max. I-FIELD SIZE

This field should only be changed when the `DEFINITION of maximum I-FIELD size` field is `user_defined`. Enter a value in bytes for the maximum I-field size. The value can range from 265 bytes to 30,729 bytes and must be multiples of 256 plus 9 but can be reduced by the system from the specified value due to factors such as the buffer size of the port and the buffer capability of the remote station. The final value should not exceed the specified value.

This option is useful when configuring to a host that has specific requirements for the I-field size but does not provide negotiation by way of XID exchanges.

TRACE Link?

This field specifies whether you want the system to save information about the activity on the link. A link trace is a sequential log of events that occur on the link that may be helpful in finding the source of a recurring error. However, performing a link trace uses processor and link time, as well as system storage. Valid values for `TRACE Link?` are as follows:

`yes` Select this value to instruct the system to save link trace information about the link. Select this value only if you experience trouble with the link and need the information to help locate the problem.

`no` Select this value to instruct the system not to save link trace information. Select this value for normal operation of the link.

Refer to SMIT Problem Determination or the **trcrpt** command in *Commands Reference* for information on getting and formatting the link trace information once it has been saved. To access SMIT Problem Determination, type `smit problem` on the AIX command line, and press Enter. Refer to the **traceson** and **tracesoff** commands in *Commands Reference* for information on starting and stopping link traces from the operating system command line.

TRACE SIZE

This field should only be changed when the `TRACE Link?` field is `yes`. Use the following information to determine the correct selection for your system:

`short` Selects the *short* level of trace reporting, which saves information on approximately 77 of the latest link activities. Each entry in the log can be up to 80 bytes long, allowing room for approximately 48 bytes of send or receive data to be saved.

`long` Selects the *long* level of trace reporting, which saves information on approximately 24 of the latest link activities. Each entry in the log can be up to 256 bytes long, allowing room for approximately 224 bytes of send or receive data to be saved.

Secondary INACTIVITY timeout

This field specifies the number of seconds that the station should wait for a transmission from the primary station. After waiting the specified period with no transmission from the primary station, the secondary station starts inactivity procedures. The value entered varies with 1) how long the primary station normally takes to respond, 2) the need to keep the link connected, and 3) the cost of keeping the secondary station waiting. If you do not know what value to use, use the default value and adjust it later, as needed.

Refer to SDLC Inactivity Procedures on page 13–80 for more information on inactivity procedures.

Local SECONDARY STATION address

This field specifies the station address for the local station. This is the address field in the SDLC data stream that identifies the secondary station. Enter a decimal value within the specified range that does not match any station address assigned to other secondary stations on the link.

Defining SDLC Primary Logical Link Characteristics

Select this option if the local station operates as an SDLC primary station. A primary station has responsibility for the data link and also issues commands to secondary stations. For example, a control station in a multipoint network is a primary station.

Use the following information to help determine the correct information for each field:

Changing the Default Characteristics

The SDLC Primary Logical Link profile example shows the information fields that comprise the SDLC logical link profile. Once the new SDLC logical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA SDLC Primary Logical Link Profile Dialog

PROFILE name	....
PHYSICAL LINK type	EIA232D
TRANSMIT window count	7
RETRANSMIT count (1–50)	10
Retransmit THRESHOLD (0–100)	10
DROP LINK on inactivity?	yes
FORCE DISCONNECT timeout (1–600 seconds)	120
DEFINITION of maximum I-FIELD size	system_defined
If user-defined, max. I-FIELD SIZE (265–30729)	265
TRACE Link?	no
If yes, TRACE SIZE	short

Primary repoll TIMEOUT (1–250, .1 second)	30
Primary repoll COUNT (3–50 repolls)	15
Primary repoll THRESHOLD (1–100%)	10
LINK type	point_to_point
If multipoint,	
Primary IDLE list timeout (30–180 seconds)	60
Primary SLOW list timeout (10–60 seconds)	60

Figure 3. Add SNA SDLC Primary Logical Link Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

PROFILE name

This field requests a name for the new logical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) for the attachments that use this SDLC logical link profile. Do not change this name without changing it throughout the entire profile.

PHYSICAL LINK type

This field designates the type of physical link being used. Use the following information to determine which selections to make:

`EIA232D` Select this type if the connection to the network uses the interface defined by Electronic Industries Association (EIA) standard EIA232D. This interface is used for directly connecting systems over short distances (less than 60 meters) or for connecting to a public switched telephone network for long distances, using a synchronous modem such as the IBM 386X synchronous modem.

`SMART MODEM` Select this type if the connection to the public switched telephone network uses a synchronous modem that employs the Smart Modem control protocol. If you select this type, specify the modem command sequence as the next field.

`X.21` Select this type if the connection to the public data network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation X.21. This interface is used to connect to the national data networks in many European countries and Japan.

`EIA422A` Select this type if the connection to the network uses the interface defined

`V.35` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.35. This interface is used to connect to the national telephone networks in many European countries and Japan.

`V.25 bis` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.25 bis. This interface is used to connect to the national telephone networks in many European countries and Japan.

TRANSMIT window count

This field specifies the number of SDLC information frames to send to the remote station before turning the line around to get a response from the remote station. The response from the remote station includes the number of frames that it received to ensure that all frames that were sent arrived at the intended destination.

The value that you supply for this field affects throughput rate on the data link. The higher the number entered for this field, the better the throughput rate. The effect on throughput rate is due to the time required to reverse the direction of data flow. Fewer changes in the direction of data flow result in more time available to transmit data. However, if the data link frequently loses frames, select a lower value for this field to save the time required to retransmit larger packages of data when an error occurs.

If you do not have a value established, use the default value. Adjust the value later to allow for data link conditions.

RETRANSMIT count

This field specifies the number of contiguous information frame bursts containing the same data that the local station retransmits before it declares a permanent transmission error. This nonproductive sequence can occur if the remote station can respond to the information frame bursts, but does not accept any of the data. For example, if this field contains a value of 10 and the remote station indicates both that:

- It accepted none of the data
- It can still receive data

then the local station transmits the data again up to 9 more times. If the remote station still does not accept the data, the local station starts inactivity procedures (see SDLC Inactivity Procedures on page 13–80).

The value entered in this field affects transmission throughput and the ability to maintain a connection with a remote station. If you specify a high value, only the worst link conditions result in disconnecting the link. However, the transmission may take a long time due to the large number of attempts to transmit the data if errors do occur. If you specify a very low value, minor transmission difficulties may result in disconnecting the link. If you do not have a value specified for the network, start with the default value and adjust it later to allow for network conditions.

Max. I-FIELD SIZE

This field should only be changed when the `DEFINITION of maximum I-FIELD size` field is `user_defined`. Enter a value in bytes for the maximum I-field size. The value can range from 265 bytes to 30,729 bytes, but can be reduced by the system from the specified value by factors such as the buffer size of the port and the buffer capability of the remote station. The final value will not exceed the specified value.

This option is useful when configuring to a host that has specific requirements for the I-field size but does not provide negotiation by way of XID exchanges.

TRACE Link?

This field specifies whether you want the system to save information about the activity on the link. A link trace is a sequential log of events that occur on the link that may be helpful in finding the source of a recurring error. However, performing a link trace uses processor and link time, as well as system storage. Valid values are the following:

`yes` Select this value to instruct the system to save link trace information about the link. Select this value only if you experience trouble with the link and need the information to help locate the problem.

`no` Select this value to instruct the system not to save link trace information. Select this value for normal operation of the link.

Refer to SMIT Problem Determination or the **trcrpt** command in *Commands Reference* for information on getting and formatting the link trace information once it has been saved. To access SMIT Problem Determination, type `smit problem` on the AIX command line, and press Enter. Refer to the **traceson** and **tracesoff** commands in *Commands Reference* for information on starting and stopping link traces from the operating system command line.

TRACE SIZE

This field should only be changed when the `TRACE Link?` field is `yes`. Use the following information to determine the correct selection for your system:

`short` Selects the *short* level of trace reporting. This level saves information on approximately 77 of the latest link activities. Each entry in the log can be up to 80 bytes long, allowing room for approximately 48 bytes of send or receive data to be saved.

`long` Selects the *long* level of trace reporting. This level saves information on approximately 24 of the latest link activities. Each entry in the log can be up to 256 bytes long, allowing room for approximately 224 bytes of send or receive data to be saved.

Primary repoll TIMEOUT

This field specifies the length of time (in tenths of a second) that the primary station should wait for a response from the secondary station. The primary station polls the secondary station and waits for a response from the secondary station.

If the secondary station does not respond within the time specified in this field, the primary station polls the secondary station again. The primary station keeps trying to get a response until the secondary station responds or until the number of tries exceeds the value specified in the `Primary repoll COUNT` field.

`multipoint` Select this option if the local system is connected to the secondary station with a multipoint link. A multipoint link connects several stations together. Data transmitted on the link is available to all stations on the link at the same time.

Primary IDLE list timeout

If the primary station has specified the `DROP LINK on inactivity?` field as `no` and discovers that a secondary station is not responding, the primary station places that secondary station on the *idle list*. This is a list of stations that are not responding. The primary station polls stations on the idle list less frequently than it does the other secondary stations to avoid tying up the network with useless polls.

The `Primary IDLE list timeout` field sets the amount of time (in seconds) that the primary station should wait between polls to stations on the idle list. When the idle list time out occurs, the primary station polls each of the secondary stations in the list one time. The secondary station is then removed from the idle list and polled at a normal rate whenever a response is received from the idle list poll. Enter a value within the indicated range to define the length of the waiting period.

Primary SLOW list timeout

When the primary station discovers that communication with a secondary station is not productive (that is, transmissions are just supervisory indications and no data is transferred), it places the secondary station on the *slow list*. This is a list of stations that are not transferring data. The primary station polls stations on the slow list less frequently than it does the other secondary stations to avoid tying up the network with useless polls.

This field sets the amount of time (in seconds) that the primary station should wait between polls to stations on the slow list. When the slow list time out occurs, the primary station polls each of the secondary stations in the list one time. The secondary station is then removed from the slow list and polled at a normal rate whenever information begins to flow between the primary and secondary stations.

Enter a value within the indicated range to define the length of the waiting period. This value should be less than the secondary station's Secondary Inactivity Timeout value to avoid having the secondary station disconnect before the poll occurs.

Defining SDLC Negotiable Logical Link Characteristics

Select this option if the local station can operate as either a primary or a secondary station, depending upon the specific remote station to which it is connected. If you select this option, the network uses LU 6.2 and connects between peer stations over a point-to-point attachment. The local secondary address at each station is automatically set to a value of 1. The connection point services at each station transfer Format 3 XIDs to determine which station is to be the primary station and which is to be the secondary station.

Use the following information to help determine the correct information for each field.

Changing the Default Characteristics

The SDLC Negotiable Logical Link profile example shows the information fields that constitute the SDLC logical link profile. Once the new SDLC logical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters. The following paragraphs supply information to help choose the values that best describe your network.

`EIA422A` Select this type if the connection to the network uses the interface defined by Electronic Industries Association (EIA) standard EIA422A. This interface is used for directly connecting systems over medium distances (less than 1.2 kilometers) or for connecting to a public switched telephone network for long distances, using a synchronous modem such as the IBM 386X synchronous modem.

`V.35` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.35. This interface is used to connect to the national telephone networks in many European countries and Japan.

`V.25 bis` Select this type if the connection to the public telephone network uses the interface defined by The International Telegraph and Telephone Consultative Committee (CCITT) recommendation V.25 bis. This interface is used to connect to the national telephone networks in many European countries and Japan.

TRANSMIT window count

This field specifies the number of SDLC information frames to send to the remote station before turning the line around to get a response from the remote station. The response from the remote station includes the number of frames that it received to ensure that all frames that were sent arrived at the intended destination.

The value that you supply for this field affects throughput rate on the data link. The higher the number entered for this field, the better the throughput rate. The effect on throughput rate is due to the time required to reverse the direction of data flow. Fewer changes in the direction of data flow result in more time available to transmit data. However, if the data link frequently loses frames, select a lower value for this field to save the time required to retransmit larger packages of data when an error occurs.

If you do not have a value established, use the default value. Adjust the value later to allow for data link conditions.

RETRANSMIT count

This field specifies the number of contiguous information frame bursts containing the same data that the local station retransmits before it declares a permanent transmission error. This nonproductive sequence can occur if the remote station can respond to the information frame bursts, but does not accept any of the data. For example, if this field contains a value of 10 and the remote station indicates both that:

- It accepted none of the data
- It can still receive data

then the local station transmits the data again up to 9 more times. If the remote station still does not accept the data, the local station starts inactivity procedures (see SDLC Inactivity Procedures on page 13–80).

DEFINITION of maximum I-FIELD SIZE

This field specifies how the maximum information field size within a link packet is determined, using the following options:

`system_defined`

This is the default selection that allows the maximum I-field size to be determined by the system by way of XID exchanges with the remote partner. The maximum value negotiated is 30,729 bytes and is reduced from that number by factors such as the buffer size of the port and the buffer capability of the remote station.

`user_defined`

This option allows you to provide your own value for the maximum I-field size. If you choose this option, select a value for the `Max.I-FIELD SIZE` field.

Max. I-FIELD SIZE

Enter a value in bytes for the maximum I-field size. The value can range from 265 bytes to 30,729 bytes, but can be reduced by the system from the specified value by factors such as the buffer size of the port and the buffer capability of the remote station. The final value will not exceed the specified value.

This option is useful when configuring to a host that has specific requirements for the I-field size but does not provide negotiation by way of XID exchanges.

TRACE Link?

This field specifies whether you want the system to save information about the activity on the link. A link trace is a sequential log of events that occur on the link that may be helpful in finding the source of a recurring error. However, performing a link trace uses processor and link time, as well as system storage. Valid values are the following:

`yes` Select this value to instruct the system to save link trace information about the link. Select this value only if you experience trouble with the link and need the information to help locate the problem.

`no` Select this value to instruct the system not to save link trace information. Select this value for normal operation of the link.

Refer to SMIT Problem Determination or the **trcrpt** command in *Commands Reference* for information on getting and formatting the link trace information once it has been saved. To access SMIT Problem Determination, type `smit problem` on the AIX command line, and press Enter. Refer to the **traceson** and **tracesoff** commands in *Commands Reference* for information on starting and stopping link traces from the operating system command line.

The value for this field varies with the secondary station's ability to respond in a timely manner. If the value is set too low, the primary station may declare that a busy, but operating, secondary station is not working. If the value is set too high, the primary station may waste a lot of time polling a secondary station that is not working. Adjust the value in this field, together with the value in the `Primary repoll TIMEOUT` field, to find a combination that works with the situation on your network.

Primary repoll THRESHOLD

This field specifies the number of repolls as a percentage of the total polls sent to the secondary station (sampled only after a block of information frames has been sent). The specified percentage equals the maximum rate of repolls allowed, above which the system declares that a temporary error has occurred. When a temporary error occurs, the system logs an entry in the error log.

Use this value to provide feedback about the performance of a particular link. If the value is very low, the error log contains information that may not indicate a link problem. If the value is very high, the error log may not contain any link information. If you do not have a value specified for the network, start with the default value and adjust it later to provide a meaningful indication of link performance in the error log.

SDLC Inactivity Procedures

SDLC inactivity procedures depend on what type of station is processing the inactivity situation and whether the operator has selected the `DROP LINK on inactivity?` option.

Primary Stations

If the station type is primary and the link is not to be dropped on inactivity, the primary station notifies its operator that the link is idle and then continues to poll the secondary station at the rate set for the `Primary IDLE list timeout` field. When the secondary station begins responding to polls from the primary station, notification is sent to the operator that the link is active again, and normal polling resumes.

If the station type is primary and the link is to be dropped on inactivity, the primary station transmits a disconnect command to the secondary station and then notifies the operator that the attachment has terminated due to inactivity. If the attachment that was terminated was the last attachment on the physical port, the physical port is also terminated and the call goes on-hook.

Secondary Stations

If the station type is secondary and the link is not to be dropped on inactivity, the secondary station notifies the operator that the link is idle and then waits in receive mode for any polls from the primary station. When the primary station begins polling the secondary, another notification is sent to the operator that the link is active again.

If the station type is secondary and the link is to be dropped on inactivity, the secondary station terminates its physical port (which goes on-hook) and then notifies the operator that the attachment has terminated due to inactivity.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t phy_eia232d ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the EIA232D physical link profile. Once the new EIA232D physical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA EIA232D Physical Link Profile Dialog

PROFILE name	....
DATALINK device name	mpq0
Serial ENCODING	NRZI
Request to send (RTS)	controlled
DTR control	DTR
Bit CLOCKING	external
If external, DATA rate select	full
If internal, TRANSMIT rate (600–38400)	1200
NETWORK type	switched
If switched, CALL type	listen
If listen,	
AUTO-LISTEN?	no
CALL-OVERRIDE?	no
ANSWER MODE	automatic

Figure 5. Add SNA EIA232D Physical Link Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

PROFILE name

This field requests a name for the new physical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) for the attachments that use this EIA232D physical link profile. Do not change this name without changing it throughout the entire profile.

Bit CLOCKING

This field indicates which piece of equipment, the modem (often referred to as the Data Communications Equipment or DCE) or the computer (Data Terminal Equipment or DTE), provides the clock signal for synchronizing data transmission. The documentation for the modem should indicate whether it provides data clocking. The options are as follows:

`external` Select this parameter if the modem provides its own data clocking. In general, synchronous modems provide do their own data clocking. If you select this parameter, you may specify the data rate for the modem.

`internal` Select this parameter if the modem does not provide data clocking. In this case, AIX SNA Services/6000 synchronizes the data before transmitting it, and when receiving, uses the incoming data to establish a clock rate. If you select this parameter, you may specify the data transmission rate.

DATA rate select

This field should only be changed when the `Bit CLOCKING` field is `external`. The selections have the following meaning:

`full` Select this value for normal (full) speed transmission of data or if the modem provides only one transmission speed.

`alternate` Select this value if the modem provides a reduced speed in addition to its normal speed and if you want to use that reduced speed for transmissions on this link. You may want to use reduced speed if frequent transmission errors occur when using full speed.

TRANSMIT rate

This field should only be changed when the `Bit CLOCKING` field is `internal`. Enter a value (in bits per second) in the indicated range for the transmission rate. This value must match the transmission rate that the remote station uses. If it does not, the two stations cannot communicate.

NETWORK type

This field indicates the type of transmission lines that implement the network, as follows:

`switched` Select this parameter if the modem is connected to the public switched telephone network. That is, the network uses ordinary telephone lines. Also, connection to a remote system requires that a telephone number be dialed first. If you select this parameter, you may use the `CALL type` field to indicate whether the local station places a call or answers a call to establish the link.

`nonswitched` Select this parameter if the network uses lines that either connect directly to the remote station or do not require a telephone number to connect to the station. A nonswitched connection can be as simple as a cable connecting two local computers (null modem) or as complex as a specially conditioned data transmission line that is leased from the telephone network and connects two systems that are many miles apart.

Defining Smart Modem Physical Link Characteristics

The Smart Modem physical link profile defines characteristics of the Smart Modem interface used to link to a remote station. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on a Smart Modem interface, you must define at least one Smart Modem physical link profile.

You can define more than one Smart Modem physical link profile, but only one set of conditions, or profile, can be active on a particular attachment. The attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) designates the physical link profile that is active on a particular attachment.

The following procedure describes how to add a Smart Modem physical link profile, using the SMIT Interface

Entering Smart Modem Physical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `SDLC.`
7. From the next SMIT menu, select `SDLC Data Link Control.`
8. From the next SMIT menu, select `SDLC Physical Link.`
9. From the next SMIT menu, select `Smart Modem Physical Link.`
10. From the next SMIT menu, select `Add a Profile.`
11. This displays the `Add SNA Smart Modem Physical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, and `Change an Alias for a Profile`.

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

DATALINK device name

This field contains the name that the local system uses for the SDLC data link device manager.

Serial ENCODING

Note: All stations on a network that communicate with each other must use the same encoding type. If you try to communicate with a remote station that uses an encoding type different from the type you are using, the remote station can not understand the transmission.

This field allows you to select the type of data encoding (NRZ or NRZI) to use for transmitting data over the data link. SDLC offers two types of data encoding, NRZ and NRZI. Neither type of encoding works for all situations. Use the following guidelines to choose the type of data encoding for the network:

- If you are attaching to an existing network, use the encoding type that the rest of the network uses.
- If you are attaching to a new network, select NRZI. This encoding type works in a majority of cases.
- If you experience problems when using one type of encoding, change the encoding type for all stations to the other type. Transmission problems that may indicate an improper encoding type include:
  - Loss of synchronization in the modem
  - Frequent disconnection
  - Random loss of data.

Request to Send (RTS)

This field selects the manner in which the link uses the **RTS** (request to send) signal while the remote and local modems are connected. Select the parameter that describes the way that the link operates, as follows:

`controlled` Select this parameter if the system activates the **RTS** signal with each transmission on the link.

`continuous` Select this parameter if the system activates the **RTS** signal when the link is established and keeps the **RTS** signal activated as long as the link is connected.

DTR control

This field determines how the **DTR** (data terminal ready) signal to the modem operates.

`CDSTL` Connect Data Set to Line: Select this value if the state of the **DTR** signal indicates an unconditional command from the data-terminal equipment (DTE) to the attached data circuit-terminating equipment (DCE) to connect or remove itself to or from the network.

`DTR` Data Terminal Ready: Select this value if the **DTR** signal merely indicates if the DTE is ready.

AUTO-LISTEN?

This field should only be changed when the CALL type field is listen. However, the AUTO-LISTEN? field is recognized if the default is changed. The AUTO-LISTEN? field determines whether the local attachment is restarted with each incoming call. The options are as follows:

yes Select this value to automatically restart the attachment after each incoming call goes on-hook.

no Select this value to prevent the attachment from restarting after an incoming call goes on-hook.

See Attachment on page 13–12 for more information about the auto-listen feature.

CALL-OVERRIDE?

This field should only be changed when the CALL type field is listen. The CALL-OVERRIDE? field determines whether an outgoing call on this link has priority over an incoming request for connection, as follows:

yes Select this value to turn off listen mode while making a call. This allows an outgoing call to a remote station to have priority over an incoming request for connection. If a request for connection is in process (but not complete) when an outgoing call is attempted, the request is dropped and the outgoing call is placed.

no Select this value to process requests for line access in the order that they occur.

ANSWER MODE

This field should only be changed when the CALL type field is listen. The ANSWER MODE field indicates whether the operator or the modem answers the phone to make a connection, as follows:

manual Select this parameter if the operator answers incoming telephone calls and then connects the station to the line.

automatic Select this parameter if the modem answers incoming telephone calls automatically.

AUTO-CALL

This field should only be changed when the CALL type field is call. Select yes if your modem and network allow an automatic dialing procedure to make the connection to the remote station. Select no to manually dial the connection to the remote station.

Connect TIMER

This field should only be changed when the CALL type field is call. The Connect TIMER field specifies the time the port waits for the call to complete before it reports an error.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t phy_x.21 ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the X.21 physical link profile. Once the new X.21 physical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA X.21 Physical Link Profile Dialog

```
PROFILE name                                        ...............
DATALINK device name                                mpq0
Serial ENCODING                                     NRZI
NETWORK type                                        switched
  If switched, CALL type                            listen
    If listen,
      AUTO-LISTEN?                                  no
      CALL-OVERRIDE?                                no
    If call,
      Number of CALL RETRIES (1–15)                 8
      DELAY between retries (1–1200, .1 sec)        300
      Signals that will cause retries:
        CPS–20  No Connection                       no
        CPS–21  Number Busy                         no
        CPS–22  Procedure Error                     no
        CPS–23  Transmission Error                  no
        CPS–61  Network Congestion                  no
        CPS–90  National Purpose 0                  no
        CPS–90  National Purpose 1                  no
        CPS–90  National Purpose 2                  no
        CPS–90  National Purpose 3                  no
        CPS–90  National Purpose 4                  no
        CPS–90  National Purpose 5                  no
        CPS–90  National Purpose 6                  no
        CPS–90  National Purpose 8                  no
        CPS–90  National Purpose 9                  no
```

Figure 7. Add SNA X.21 Physical Link Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

CALL type

This field should only be changed when the `NETWORK type` field is `switched`. This field indicates whether the local station initiates a connection, or receives requests for a connection from another station. Select one of the following values that describes how the local station operates. You must select the same value for this field that you selected for the same field in the corresponding attachment profile (see Defining SDLC Attachment Characteristics on page 13–56). The options are as follows:

`call` Select this value to indicate that the local station initiates a connection by calling another station.

`listen` Select this value to indicate that the local station does not initiate a connection, but waits for a remote station to make a connection with it. If this value is selected, you may change the default values for the `AUTO-LISTEN?`, `CALL-OVERRIDE?`, and `ANSWER MODE` fields.

AUTO-LISTEN?

This field should only be changed when the `CALL type` field is `listen`. The `AUTO-LISTEN?` field determines whether the local attachment is restarted with each incoming call. See Attachment on page 13–12 for more information about the auto-listen feature.

`yes` Select this value to automatically restart the attachment after each incoming call goes on-hook.

`no` Select this value to prevent the attachment from restarting after an incoming call goes on-hook.

CALL-OVERRIDE?

This field should only be changed when the `CALL type` field is `listen`. The `CALL-OVERRIDE?` field determines whether an outgoing call on this link has priority over an incoming request for connection, as follows:

`yes` Select this value to turn off listen mode while making a call. This allows an outgoing call to a remote station to have priority over an incoming request for connection. If a request for connection is in process (but not complete) when an outgoing call is attempted, the request is dropped and the outgoing call is placed.

`no` Select this value to process requests for line access in the order that they occur.

Number of CALL RETRIES

This field should only be changed when the `CALL type` field is `call`. This field determines the number of times that the system tries to complete a call unsuccessfully before abandoning the call. This field may be specified by the network to which you are connected. Check the network regulations before entering a number in this field. If the network does not specify a value, select the default value.

DELAY between retries

This field should only be changed when the `CALL type` field is `call`. This field determines the number of seconds that the system waits before it tries again to complete a previously unsuccessful call. This field may be specified by the network to which you are connected. Check the network regulations before entering a number in this field. If the network does not specify a value, select the default value.

Defining EIA422A Physical Link Characteristics

The EIA422A physical link profile defines characteristics of the EIA422A network. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on an EIA422A network, you must define at least one EIA422A physical link profile.

You can define more than one EIA422A physical link profile, but only one set of conditions, or profile, can be active on a particular attachment. The attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) designates the physical link profile that is active on a particular attachment.

The following procedure describes how to add a EIA422A physical link profile, using the SMIT Interface

Entering EIA422A Physical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `SDLC.`
7. From the next SMIT menu, select `SDLC Data Link Control.`
8. From the next SMIT menu, select `SDLC Physical Link.`
9. From the next SMIT menu, select `EIA422A Physical Link.`
10. From the next SMIT menu, select `Add a Profile.`
11. This displays the `Add SNA EIA422A Physical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Serial ENCODING

Note: All stations on a network that communicate with each other must use the same encoding type. If you try to communicate with a remote station that uses an encoding type different from the type you are using, the remote station can not understand the transmission.

This field allows you to select the type of data encoding (NRZ or NRZI) to use for transmitting data over the data link. SDLC offers two types of data encoding, NRZ and NRZI. Neither type of encoding works for all situations. Use the following guidelines to choose the type of data encoding for the network:

- If you are attaching to an existing network, use the encoding type that the rest of the network uses.
- If you are attaching to a new network, select NRZI. This encoding type works in a majority of cases.
- If you experience problems when using one type of encoding, change the encoding type for all stations to the other type. Transmission problems that may indicate an improper encoding type include:
  - Loss of synchronization in the modem
  - Frequent disconnection
  - Random loss of data.

Request to send (RTS)

This field selects the manner in which the link uses the **RTS** (request to send) signal while the remote and local modems are connected. Select the parameter that describes the way that the link operates, as follows:

`controlled` Select this parameter if the system activates the **RTS** signal with each transmission on the link.

`continuous` Select this parameter if the system activates the **RTS** signal when the link is established and keeps the **RTS** signal activated as long as the link is connected.

DTR control

This field determines how the **DTR** (data terminal ready) signal to the modem operates, as follows:

`CDSTL` Connect Data Set to Line: Select this value if the state of the **DTR** signal indicates an unconditional command from the data terminal equipment (DTE) to the attached data circuit-terminating equipment (DCE) to connect or remove itself to or from the network.

`DTR` Data Terminal Ready: Select this value if the **DTR** signal merely indicates if the DTE is ready or not.

Defining V.25 bis Physical Link Characteristics

The V.25 bis physical link profile defines characteristics of the Electronic Industries Association (EIA) standard V.25 bis interface, used for the link to a remote station. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on a V.25 bis interface, you must define at least one V.25 bis physical link profile.

You can define more than one V.25 bis physical link profile, but only one set of conditions, or profile, can be active on a particular attachment. The attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) designates the physical link profile that is active on a particular attachment.

The following procedure describes how to add a V.25 bis physical link profile, using the SMIT Interface

Entering V.25 bis Physical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `SDLC.`
7. From the next SMIT menu, select `SDLC Data Link Control.`
8. From the next SMIT menu, select `SDLC Physical Link.`
9. From the next SMIT menu, select `V.25 bis Physical Link.`
10. From the next SMIT menu, select `Add a Profile.`
11. This displays the `Add SNA V.25 bis Physical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

DATALINK device name

This field contains the name that the local system uses for the SDLC data link device manager.

Serial ENCODING

Note: All stations on a network that communicate with each other must use the same encoding type. If you try to communicate with a remote station that uses an encoding type different from the type you are using, the remote station will not understand the transmission.

This field allows you to select the type of data encoding (NRZ or NRZI) to use for transmitting data over the data link. SDLC offers two types of data encoding, NRZ and NRZI. Neither type of encoding works for all situations. Use the following guidelines to choose the type of data encoding for the network:

- If you are attaching to an existing network, use the encoding type that the rest of the network uses.
- If you are attaching to a new network, select NRZI. This encoding type works in a majority of cases.
- If you experience problems when using one type of encoding, change the encoding type for all stations to the other type. Transmission problems that may indicate an improper encoding type include:
  - Loss of synchronization in the modem
  - Frequent disconnection
  - Random loss of data.

Request to send (RTS)

This field selects the manner in which the link uses the **RTS** (request to send) signal while the remote and local modems are connected. Select the parameter that describes the way that the link operates, as follows:

`controlled` Select this parameter if the system activates the **RTS** signal with each transmission on the link.

`continuous` Select this parameter if the system activates the **RTS** signal when the link is established and keeps the **RTS** signal activated as long as the link is connected.

DTR control

This field determines how the **DTR** (data terminal ready) signal to the modem operates, as follows.

`CDSTL` Connect Data Set to Line: Select this value if the state of the **DTR** signal indicates an unconditional command from the DTE (data terminal equipment) to the attached DCE (data circuit-terminating equipment) to connect or remove itself to or from the network.

`DTR` Data Terminal Ready: Select this value if the **DTR** signal merely indicates if the DTE is ready or not.

`no` Select this value to prevent the attachment from restarting after an incoming call goes on-hook.

CALL-OVERRIDE?

This field should only be changed if the `CALL type` field is `listen`. The `CALL-OVERRIDE?` field determines whether an outgoing call on this link has priority over an incoming request for connection, as follows:

`yes` Select this value to turn off listen mode while making a call. This allows an outgoing call to a remote station to have priority over an incoming request for connection. If a request for connection is in process (but not complete) when an outgoing call is attempted, the request is dropped and the outgoing call is placed.

`no` Select this value to process requests for line access in the order that they occur.

ANSWER MODE

This field should only be changed if the `CALL type` field is `listen`. The `ANSWER MODE` field indicates whether the operator or the modem answers the phone to make a connection,as follows:

`manual` Select this parameter if the operator answers incoming telephone calls and then connects the station to the line.

`automatic` Select this parameter if the modem answers incoming telephone calls automatically.

Connect TIMER

This field should only be changed when the `CALL type` field is `call`. The `Connect TIMER` field specifies the time the port waits for the call to complete before it reports an error.

Transmit DELAY

This field should only be changed when the `CALL type` field is `call`. The `Transmit DELAY` field specifies the time delay between driving the data terminal ready (**DTR**) signal and sending the dial data to the modem.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t phy_v.35 ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the V.35 physical link profile. Once the new V.35 physical link profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA V.35 Physical Link Profile Dialog

PROFILE name	....
DATALINK device name	mpq0
Serial ENCODING	NRZI
Request to send (RTS)	controlled
DTR control	DTR
Bit CLOCKING	external
If external, DATA rate select	full
If internal, TRANSMIT rate (600–38400)	19200

Figure 10. Add SNA V.35 Physical Link Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

PROFILE name

This field requests a name for the new physical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining SDLC Attachment Characteristics on page 13–56) for the attachments that use this V.35 physical link profile. Do not change this name without changing it throughout the entire profile.

DATALINK device name

This field contains the name that the local system uses for the SDLC data link device manager.

Bit CLOCKING

This field indicates which piece of equipment, the modem (often referred to as the Data Communications Equipment or DCE) or the computer (Data Terminal Equipment or DTE), provides the clock signal for synchronizing data transmission. The documentation for the modem should indicate whether it provides data clocking. The options are as follows:

`external` Select this parameter if the modem provides its own data clocking. In general, synchronous modems provide do their own data clocking. If you select this parameter, you may specify the data rate for the modem.

`internal` Select this parameter if the modem does not provide data clocking. In this case, AIX SNA Services/6000 synchronizes the data before transmitting it, and when receiving, uses the incoming data to establish a clock rate. If you select this parameter, you may specify the data transmission rate.

DATA rate select

This field should only be changed when the `Bit CLOCKING` field is `external`. The selections have the following meaning:

`full` Select this value for normal (full) speed transmission of data or if the modem provides only one transmission speed.

`alternate` Select this value if the modem provides a reduced speed in addition to its normal speed and if you want to use that reduced speed for transmissions on this link. You may want to use reduced speed if frequent transmission errors occur when using full speed.

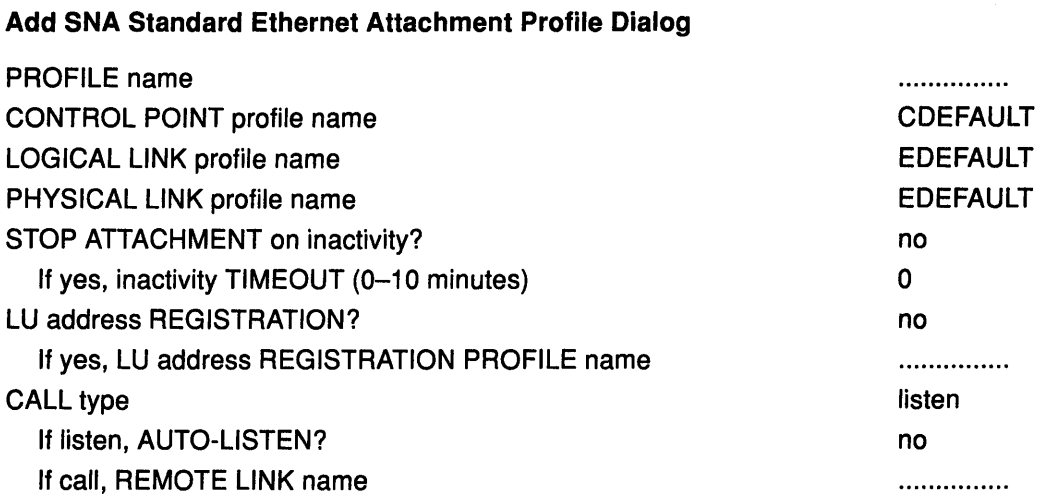
TRANSMIT rate

This field should only be changed when the `Bit CLOCKING` field is `internal`. Enter a value (in bits per second) in the indicated range for the transmission rate. This value must match the transmission rate that the remote station uses. If it does not, the two stations cannot communicate.

Changing the Default Characteristics

The following profile example shows the information fields that constitute the Standard Ethernet attachment profile. Once the new attachment profile appears on the screen, you must enter some additional information to correlate the attachment with the other profiles that describe it. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.



Add SNA Standard Ethernet Attachment Profile Dialog

PROFILE name	....
CONTROL POINT profile name	CDEFAULT
LOGICAL LINK profile name	EDEFAULT
PHYSICAL LINK profile name	EDEFAULT
STOP ATTACHMENT on inactivity?	no
If yes, inactivity TIMEOUT (0–10 minutes)	0
LU address REGISTRATION?	no
If yes, LU address REGISTRATION PROFILE name	....
CALL type	listen
If listen, AUTO-LISTEN?	no
If call, REMOTE LINK name	....

Figure 11. Add SNA Standard Ethernet Attachment Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests a name for the new attachment profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

CONTROL POINT profile name

This field provides the name of the control point profile that defines the node ID of the physical unit associated with this attachment. You must create a control point profile that has the name that you provide in this field. Refer to Defining Physical Unit Characteristics on page 13–215 for information about creating this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

LOGICAL LINK profile name

This field provides the name of the logical link profile that defines the characteristics of the data link protocol that implements the network. You must create a logical link profile having the name you provide in this field. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on choosing a name for this field.

Refer to Defining Standard Ethernet Logical Link Characteristics on page 13–113 for information about creating the logical link profile.

CALL type

This field indicates whether the local station initiates a connection or receives requests for a connection from another station. Select one of the following values that describes how the local station operates:

`call`	Select this value to indicate that the local station initiates a connection by calling another station.
`listen`	Select this value to indicate that the local station does not initiate a connection but waits for a remote station to make a connection with it.

AUTO-LISTEN?

This field should only be changed if `CALL type` is `listen`. The `listen` parameter determines whether the local attachment is restarted with each incoming call. The options are as follows:

`yes`	Select this value to restart the attachment with an incoming call.
`no`	Select this value to prevent the attachment from restarting with an incoming call.

REMOTE LINK name

This field should only be changed if the `CALL type` field is `call`. Enter the unique link name for the remote station. This name is entered in the `LOCAL LINK name` field of the Ethernet physical link profile on the remote system. You must contact the remote station to find out this parameter. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on selecting a name for this field.

Refer to Defining Standard Ethernet Physical Link Characteristics on page 13–119 for more information on the physical link profile.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t log_ethnet ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the Standard Ethernet logical link profile. Once the new Standard Ethernet logical link profile appears on the screen, you can accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA Standard Ethernet Logical Link Profile Dialog

PROFILE name	....
TRANSMIT window count (1–127)	10
RETRANSMIT count (1–30)	8
RECEIVE window count (1–127)	127
DROP LINK on inactivity?	yes
INACTIVITY timeout (1–120 seconds)	48
RESPONSE timeout (1–40, 500 msec intervals)	2
ACKNOWLEDGE timeout (1–40, 500 msec intervals)	1
FORCE DISCONNECT timeout (1–600 seconds)	120
DEFINITION of maximum I-FIELD size	system_defined
If user-defined, max. I-FIELD SIZE (265–30729)	1417
TRACE Link?	no
If yes, TRACE SIZE	short

Figure 12. Add SNA Standard Ethernet Logical Link Profile Dialog

The following paragraphs supply information to help you choose the values that best describe your network.

PROFILE name

This field requests a name for the new logical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining Standard Ethernet Attachment Characteristics on page 13–109) for the attachments that use this logical link profile. Do not change this name without changing it throughout the entire profile.

TRANSMIT window count

This field specifies the number of information packets to send to the remote station before waiting for a response from the remote station. The response from the remote station includes the number of packets that it received to ensure that all packets that were sent arrived at the intended destination.

However, restarting the link takes time to resolve the profiles associated with a connection, and you may want to keep the link active to avoid the delay. Unless keeping the link open is very important, select `yes` and enter a value for the `INACTIVITY timeout` field. Refer to Standard Ethernet Inactivity Procedures on page 13–118 for more Ethernet inactivity information. Available options are as follows:

`yes`	Select this value to drop the link after a period of inactivity.
`no`	Select this value to receive notification of inactivity and then remain active regardless of how long the link remains idle.

INACTIVITY timeout

This field determines the number of seconds to wait for a valid receive packet from the remote station. Once this timeout expires, the remote station is polled with an appropriate command at the rate specified in the `RESPONSE timeout` field. If the remote station does not respond to the polls after the number of polls specified in the `RETRANSMIT count` field is completed, the local station starts inactivity procedures.

Refer to Standard Ethernet Inactivity Procedures on page 13–118 for more Ethernet inactivity information.

RESPONSE timeout

This field specifies the number of seconds that the station should wait for a required response from the remote station. After waiting for the specified period with no response from the remote station, the local station retransmits the command up to the number of retries specified in the `RETRANSMIT count` field.

Enter a value for the number of 500-millisecond intervals the local station should wait for a response. The value entered varies with 1) how long the remote station normally takes to respond, 2) the need to keep the link connected, and 3) the cost of keeping the local station waiting.

This value can range from 1 to 40 (500-millisecond) intervals. If you do not know what value to use, use the default value, 2, and adjust it as needed.

ACKNOWLEDGE timeout

This field specifies the number of seconds that the local station should wait before sending an acknowledgment to the remote station after receiving data. After waiting for the specified period with no command poll from the remote station, the local station sends the acknowledgment.

Enter a value that is the number of 500-millisecond intervals to wait before sending an acknowledgment. Waiting uses buffer space at the remote station. The remote station holds the transmitted data until the acknowledgment is received.

This value can range from 1 to 40 (500-millisecond) intervals. If you do not know what value to use, use the default value, 1, and adjust it as needed.

FORCE DISCONNECT timeout

This field specifies the number of seconds that the system should wait after requesting a disconnect (`DISC`) from the link before the system forces the disconnect. The value that you choose varies with the network. Choose a value that is reasonable within the abilities of the network to respond to a disconnect without tying up local system resources for an extended period of time waiting for the disconnect. This value can range from 1 to 600 seconds. If you do not know what value to use, select the default value, 120 seconds, adjusting it as necessary to respond to actual network performance.

`long` Selects the *long* level of trace reporting, which saves information on approximately 24 of the latest link activities. Each entry in the log can be up to 256 bytes long, allowing room for approximately 224 bytes of send or receive data to be saved.

Standard Ethernet Inactivity Procedures

Standard Ethernet inactivity procedures depend on whether the operator has selected `yes` for the `DROP LINK on inactivity?` field.

If `no` is selected for that field, the local station notifies its operator that the link is idle and then continues to poll the remote station at the rate set in the `RESPONSE timeout` field. When the remote station begins responding to the local station's polls, another notification is sent to the operator that the link is active again. Normal polling resumes.

If `yes` is selected for that field, the local station notifies its operator that the attachment has terminated due to inactivity. If the attachment that is terminated is the last attachment on the physical port, the physical port is also terminated.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t phy_ethnet ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the Standard Ethernet physical link profile. Once the new Standard Ethernet physical link profiles appear on the screen, you can accept the default values for the fields. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA Standard Ethernet Physical Link Profile Dialog

PROFILE name	....
DATALINK device name	ent0
LOCAL LINK name	....
Maximum number of LOGICAL LINKS (1–255)	32
Local SAP address (0x04 – 0xEC)	04

Figure 13. Add SNA Standard Ethernet Physical Link Profile Dialog

The following paragraphs supply information to help you choose the values that best describe your network.

PROFILE name

This field requests a name for the new physical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining Standard Ethernet Attachment Characteristics on page 13–109) for the attachments that use the Standard Ethernet physical link profile. Do not change this name without changing it in the other places where it is used.

DATALINK device name

This field contains the name that the local system uses for the Standard Ethernet data link device manager.

LOCAL LINK name

This field identifies the local node to other nodes on the network. The name entered here cannot be assigned to any other node on the network. Select any string of characters for this field that has meaning to you within the restrictions described in AIX SNA Services/6000 Naming Requirements on page 13–25. This name cannot be the same as the name specified for the Remote Link Name in the associated attachment profile.

When you start the connection that uses this profile, AIX SNA Services/6000 checks the network to ensure that the name you specify here is not already in use by some other node. If the name is being used, AIX SNA Services/6000 generates an error message. You must then change this field to specify a name that is not being used.

Defining IEEE 802.3 Ethernet Attachment Characteristics

The attachment profile contains fields that both associate other defined profiles with the attachment of the LU to the network and define the type of network being used. AIX SNA Services/6000 uses the information in this profile to open an attachment to the described network through a hardware adapter. Each attachment to a network must have an attachment profile defined for it.

The following procedure describes how to add an IEEE 802.3 Ethernet attachment profile, using the SMIT Interface

Entering IEEE 802.3 Ethernet Attachment Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `802.3 Ethernet.`
7. From the next SMIT menu, select `802.3 Ethernet Attachment.`
8. From the next SMIT menu, select `Add a Profile.`
9. This displays the `Add SNA 802.3 Ethernet Attachment Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, `Change an Alias for a Profile`, and `Change Generic LU Address Registration.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t attachment –w 802.3 ProfileName
```

PHYSICAL LINK profile name

This field provides the name of the physical link profile that defines the characteristics of the physical port for the network. You must create a physical link profile having the name you provide in this field. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on choosing a name for this field.

Refer to Defining IEEE 802.3 Ethernet Physical Link Characteristics on page 13–132 for information about creating the physical link profile.

STOP ATTACHMENT on inactivity?

This field specifies whether AIX SNA Services/6000 should stop the attachment if no connections are active on the attachment for a specified period of time. This choice affects system performance because each open attachment uses system resources. If the attachment is idle, the resources may be used more efficiently elsewhere.

However, restarting the attachment takes time to resolve the profiles associated with a connection, and you may want to keep the attachment active to avoid the delay. In addition, if the link is not terminated when the stop occurs, cleanup procedures that require interaction with other nodes may not be completed. If this happens, put the link back into an active state for a normal stop to occur.

In general, unless keeping the attachment open is very important, select `yes` and choose a value for the time-out period.

`yes` Select this value to stop the attachment after a period of inactivity. If you select this value, you must specify the length of the inactivity period in the following field.

`no` Select this value to remain attached to the remote node regardless of how long the connection remains idle.

Inactivity TIMEOUT

This field should only be changed and is required if the `STOP ATTACHMENT on inactivity?` field is `yes`. Enter a value in the specified range (0 to 10) for the number of minutes to wait before stopping the attachment. The actual waiting period is within 30 seconds of the specified number of minutes.

LU address REGISTRATION?

This field specifies whether LU addresses is registered for use by the attachment with the generic SNA application.

`yes` Select this value if you are using generic SNA and LU addresses are registered. Enter the name of the profile containing the list of registered LU addresses in the `LU address REGISTRATION PROFILE name` field.

`no` Select this value if you are not using generic SNA and do not need registered LU addresses.

LU address REGISTRATION PROFILE name

This field should only be changed if the `LU address REGISTRATION?` field is `yes`. This field provides the name of the LU address registration profile name that contains a line of LU addresses to be registered for use by this attachment with generic SNA.

Defining IEEE 802.3 Ethernet Logical Link Characteristics

The 802.3 Ethernet logical link profile defines characteristics of the 802.3 Ethernet network. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on an 802.3 Ethernet network, you must define at least one 802.3 Ethernet logical link profile.

You can define more than one 802.3 Ethernet logical link profile, but only one set of conditions, or profile, can be active on a particular attachment. The attachment profile (see Defining IEEE 802.3 Ethernet Attachment Characteristics on page 13–122) designates the logical link profile that is active on a particular attachment.

The following procedure describes how to add an IEEE 802.3 Ethernet logical link profile, using the SMIT Interface

Entering IEEE 802.3 Ethernet Logical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `802.3 Ethernet.`
7. From the next SMIT menu, select `802.3 Ethernet Data Link Control.`
8. From the next SMIT menu, select `802.3 Ethernet Logical Link.`
9. From the next SMIT menu, select `Add a Profile.`
10. This displays the `Add SNA 802.3 Ethernet Logical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

The value that you supply for this field affects data reliability and throughput rate on the data link. The higher the number entered for this field, the better the throughput rate. The effect on throughput rate is due to the time spent waiting for a response. Fewer received responses result in more time available to transmit data. However, if the data link frequently loses information, select a lower value for this field to save the time required to retransmit larger packages of data when an error occurs.

If you do not have a value established, use the default value of 10. Adjust the value later to allow for data link conditions.

RETRANSMIT count

This field specifies the number of times the local station should poll the remote station unsuccessfully before marking the remote station as not working. Polling occurs on all unnumbered and supervisory commands, and is also started whenever an inactivity time out has occurred. Each poll command is transmitted to the remote at an interval specified in the `RESPONSE timeout` field. If the remote station does not respond within the number of polls specified, the local station starts inactivity procedures (see IEEE 802.3 Ethernet Inactivity Procedures on page 13–131).

The value for this field varies with the remote station's ability to respond in a timely manner. If the value is too low, the local station may declare that a busy, but operating, remote station is not working. If the value is too high, the local station may waste a lot of time polling a remote station that is not working. Adjust the value of this field and the values in the `INACTIVITY timeout` and `RESPONSE timeout` fields to find a combination that works on your network. The default value is 8.

RECEIVE window count

This field specifies the number of information packets to receive from the remote station before sending an acknowledgment to the remote station. The acknowledgment includes the number of packets received, which ensures that all packets sent arrived at the intended destination.

The value that you supply for this field affects both how storage is used on the remote system and the throughput rate on the data link. A higher value for this field increases the throughput rate, which in turn increases the storage required on the remote system for buffering the transmit data. Throughput rate is effected because acknowledgments are not productive transmissions (do not carry data).

As the value for this field increases, fewer acknowledgments are sent in a specific period of time. Therefore, more data can be sent. Since, the remote station must hold its data for possible retransmission until an acknowledgment is received from the local station, the remote station's storage requirements increases as a result.

This value can range from 1 to 127. If you do not have a value established, use the default value of 127. Adjust the value later to allow for data link conditions.

DROP LINK on inactivity?

This field specifies whether AIX SNA Services/6000 should drop the link if no response is received from the remote station for a specified period of time. This choice affects system performance because each open link uses system resources. If the link is idle, the resources may be used more efficiently elsewhere.

FORCE DISCONNECT timeout

This field specifies the number of seconds that the system should wait after requesting a disconnect (DISC) from the link before the system forces the disconnect. The value that you choose varies with the network. Choose a value that is reasonable within the abilities of the network to respond to a disconnect without tying up local system resources for an extended period of time waiting for the disconnect. This value can range from 1 to 600 seconds. If you do not know what value to use, select the default value, 120 seconds, adjusting it as necessary to respond to actual network performance.

DEFINITION of maximum I-FIELD size

This field specifies how the maximum information field size within a link packet is determined, using the following options:

`system_defined`
: This is the default selection that allows the maximum I-field size to be determined by the system by way of XID exchanges with the remote partner. The maximum value negotiated is 30,729 bytes, which may be reduced by factors such as the buffer size of the port and the buffer capability of the remote station.

`user_defined`
: This option allows you to provide your own value, using the following field, for the maximum I-field size.

Max. I-FIELD SIZE

This field should only be changed if the `DEFINITION of maximum I-FIELD size` field is `user_defined`. Enter a value in bytes for the maximum I-field size. The value can range from 265 bytes to 30,729 bytes, but can be reduced by the system from the specified value due to factors such as the buffer size of the port and the buffer capability of the remote station. The final value will not exceed the specified value.

This option is useful when configuring to a host that has specific requirements for the I-field size but does not provide negotiation by way of XID exchanges.

TRACE Link?

This field specifies whether you want the system to save information about the activity on the link. A link trace is a sequential log of events that occur on the link that may be helpful in finding the source of a recurring error. However, performing a link trace uses processor and link time, as well as system storage. Valid values are the following:

`yes`
: Select this value to instruct the system to save link trace information about the link. Select this value only if you experience trouble with the link and need the information to help locate the problem.

`no`
: Select this value to instruct the system not to save link trace information. Select this value for normal operation of the link.

Refer to SMIT Problem Determination or the **trcrpt** command in *Commands Reference* for information on getting and formatting the link trace information once it has been saved. To access SMIT Problem Determination, type `smit problem` on the AIX command line, and press Enter. Refer to the **traceson** and **tracesoff** commands in *Commands Reference* for information on starting and stopping link traces from the operating system command line.

Defining IEEE 802.3 Ethernet Physical Link Characteristics

The IEEE 802.3 Ethernet physical link profile defines characteristics of an 802.3 Ethernet link. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on an 802.3 Ethernet network, you must define at least one 802.3 Ethernet physical link profile.

You can define more than one 802.3 Ethernet physical link profile, but only one set of conditions, or profile, can be active on a particular port. The attachment profile (see Defining IEEE 802.3 Ethernet Attachment Characteristics on page 13–122) designates the physical link profile that is active on a particular attachment.

The following procedure describes how to add an IEEE 802.3 Ethernet physical link profile, using the SMIT Interface

Entering IEEE 802.3 Ethernet Physical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `802.3 Ethernet.`
7. From the next SMIT menu, select `802.3 Ethernet Data Link Control.`
8. From the next SMIT menu, select `802.3 Ethernet Physical Link.`
9. From the next SMIT menu, select `Add a Profile.`
10. This displays the `Add SNA 802.3 Ethernet Physical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Maximum number of LOGICAL LINKS

This field defines the maximum number of logical links that SNA can activate on this port at one time. The value for this field affects the local system's performance and response time for exchanges with remote systems, as well as its ability to establish a logical link with a remote site.

Lower values allow non-SNA logical links to share the same data link and adapter, but restrict the ability of SNA to establish an attachment. Higher values provide SNA with better access to the network, but restrict non-SNA attachments. The default value is 32.

Local SAP address

Enter the service access point (SAP) address for the transaction program on the local system. This address is a hexadecimal value in the range 04 to EC. It does not need to be unique within the network. If you are setting up a new network, assign a value of 04 as the SAP address for each SNA Service in the network.

Changing the Default Characteristics

The following profile example shows the information fields that constitute the attachment profile. Once the new attachment profile appears on the screen, you must enter some additional information to correlate the attachment with the other profiles that describe it. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press the F7 (`Enter=Do`) key to save the new profile with the modified parameters.

Add SNA Token-Ring Attachment Profile Dialog

```
PROFILE name                                                  ...............
CONTROL POINT profile name                                    CDEFAULT
LOGICAL LINK profile name                                     TDEFAULT
PHYSICAL LINK profile name                                    TDEFAULT
STOP ATTACHMENT on inactivity?                                no
  If yes, inactivity TIMEOUT (0–10 minutes)                   0
LU address REGISTRATION?                                      no
  If yes, LU address REGISTRATION PROFILE name                ...............
CALL Type                                                     listen
  If listen,
    AUTO-LISTEN?                                       no
    MINIMUM SAP Address (0x04 – 0xEC)                         04
    MAXIMUM SAP Address (0x04 – 0xEC)                         EC
  If call, ACCESS ROUTING                                     link_name
    If link-name, REMOTE LINK name                            ...............
    If link-address,
      Remote LINK address                                     0
      Remote SAP address (0x04 – 0xEC)                        04
```

Figure 17. Add SNA Token-Ring Attachment Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests a name for the new attachment profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

CONTROL POINT profile name

This field provides the name of the control point profile that defines the node ID of the physical unit associated with this attachment. You must create a control point profile that has the name that you provide in this field. Refer to Defining Physical Unit Characteristics on page 13–215 for information about creating this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

LU address REGISTRATION PROFILE name

This field should only be changed if the `LU address REGISTRATION?` field is `yes`. This field provides the name of the LU address registration profile name that contains a list of LU addresses to be registered for use by this attachment with generic SNA.

CALL type

This field indicates whether the local station initiates a connection or receives requests for a connection from another station. Select one of the following values that describes how the local station operates:

`call` Select this value to indicate that the local station initiates a connection by calling another station. If you select this option, you may request the access routing:

`listen` Select this value to indicate that the local station does not initiate a connection, but waits for a remote station to make a connection with it.

AUTO-LISTEN?

This field should only be changed if the `CALL type` field is `listen`. The `AUTO-LISTEN?` field determines whether the local attachment is restarted with each incoming call. The options are as follows:

`yes` Select this value to restart the attachment with an incoming call.

`no` Select this value to prevent the attachment from restarting with an incoming call.

MINIMUM SAP Address

This field defines the minimum service access point (SAP) address for the transaction program on the remote system. The minimum address, expressed in hexadecimal, must be equal to or less than the Local SAP on the token-ring physical link profile for the remote station. The call cannot be completed if the Local SAP is not in range.

MAXIMUM SAP Address

This field defines the maximum service access point (SAP) address for the transaction program on the remote system. The maximum address, expressed in hexadecimal, must be equal to or greater than the Local SAP on the token-ring physical link profile for the remote station. The call cannot be completed if the Local SAP is not in range.

The maximum SAP address does not need to be unique within the network. If you are setting up a new network, assign a value of EC for the maximum SAP address for each SNA Service in the network.

ACCESS ROUTING

This field should only be changed if the `CALL type` field is `call`. The `ACCESS ROUTING` field determines whether the remote station is contacted by using the name of the remote station or by using the address of the remote station.

`link_name` Select this value if the remote station is contacted by name. If this value is selected, enter the name into the `REMOTE LINK name` field.

`link_address`

Select this value if the remote station is contacted by address. If this value is selected, enter the address into the `Remote SAP Address` and `Remote LINK address` field.

Defining Token-Ring Logical Link Characteristics

The Token-Ring logical link profile defines characteristics of the Token-Ring network. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on a Token-Ring network, you must define at least one Token-Ring logical link profile.

You can define more than one Token-Ring logical link profile, but only one set of conditions, or profile, can be active on a particular attachment. The attachment profile (see Defining Token-Ring Attachment Characteristics on page 13–135) designates the logical link profile that is active on a particular attachment.

The following procedure describes how to add a Token-Ring logical link profile, using the SMIT Interface

Entering Token-Ring Logical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `Token Ring.`
7. From the next SMIT menu, select `Token Ring Data Link Control.`
8. From the next SMIT menu, select `Token Ring Logical Link.`
9. From the next SMIT menu, select `Add a Profile.`
10. This displays the `Add SNA Token Ring Logical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

TRANSMIT window count

This field specifies the number of information packets to send to the remote station before waiting for a response from the remote station. The response from the remote station includes the number of packets that it received to ensure that all packets that were sent arrived at the intended destination.

The value that you supply for this field affects data reliability and throughput rate on the data link. The higher the number entered for this field, the better the throughput rate. The effect on throughput rate is due to the time spent waiting for a response. Fewer received responses result in more time available to transmit data. However, if the data link frequently loses information, select a lower value for this field to save the time required to retransmit larger packages of data when an error occurs.

The value can range from 1 to 127. If you do not have a value established, use the default value of 10. Adjust the value later to allow for data link conditions.

DYNAMIC window increment

When network congestion occurs (the receive buffers at a bridge are full), the local transmit window count drops to 1. The dynamic window field specifies the number of consecutive packets that must be acknowledged before the local transmit window count can be raised. This gradually increases the traffic once the congestion has cleared. The value can range from 1 to 127. If you do not have a value established, use the default value of 1.

RETRANSMIT count

This field specifies the number of times the local station should poll the remote station unsuccessfully before marking the remote station as not working. Polling occurs on all unnumbered and supervisory commands, and is also started whenever an inactivity time out has occurred. Each poll command is transmitted to the remote at an interval specified in the `RESPONSE timeout` field. If the remote station does not respond within the number of polls specified, the local station starts inactivity procedures (see Token-Ring Inactivity Procedures on page 13–145).

The value for this field varies with the remote station's ability to respond in a timely manner. If the value is too low, the local station may declare that a busy, but operating, remote station is not working. If the value is too high, the local station may waste a lot of time polling a remote station that is not working. Adjust the value of this field and the values in the `INACTIVITY timeout` and `RESPONSE timeout` fields to find a combination that works on your network. The default value is 8.

RECEIVE window count

This field specifies the number of information packets to receive from the remote station before sending an acknowledgment to the remote station. The acknowledgment includes the number of packets received, which ensures that all packets sent arrived at the intended destination.

The value that you supply for this field affects both how storage is used on the remote system and the throughput rate on the data link. A higher value for this field increases the throughput rate, which in turn increases the storage required on the remote system for buffering the transmit data. Throughput rate is affected because acknowledgments are not productive transmissions (do not carry data).

ACKNOWLEDGE timeout

This field specifies the number of seconds that the local station should wait before sending an acknowledgment to the remote station after receiving data. After waiting for the specified period with no command poll from the remote station, the local station sends the acknowledgment.

Enter a value that is the number of 500-millisecond intervals to wait before sending an acknowledgment. Waiting uses buffer space at the remote station. The remote station holds the transmitted data until the acknowledgment is received. This value can range from 1 to 40 500-millisecond intervals. If you do not know what value to use, use the default value of 1, and adjust it as needed.

FORCE DISCONNECT timeout

This field specifies the number of seconds that the system should wait after requesting a disconnect (DISC) from the link before the system forces the disconnect. The value that you choose varies with the network. Choose a value that is reasonable within the abilities of the network to respond to a disconnect without tying up local system resources for an extended period of time waiting for the disconnect. This value can range from 1 to 600 seconds. If you do not know what value to use, select the default value of 120 seconds, adjust it as necessary to respond to actual network performance.

DEFINITION of maximum I-FIELD size

This field specifies how the maximum information field size within a link packet is determined, by using the following options:

`system_defined`

This is the default selection that allows the maximum I-field size to be determined by the system by way of XID exchanges with the remote partner. The maximum value negotiated is 30,729 bytes, which may be reduced by factors such as the buffer size of the port and the buffer capability of the remote station.

`user_defined`

This option allows you to provide your own value, using the following field, for the maximum I-field size.

Max. I-FIELD SIZE

This field should only be changed if the `DEFINITION of maximum I-FIELD size` field is `user_defined`. Enter a value in bytes for the maximum I-field size. The value can range from 265 bytes to 30,729 bytes, but can be reduced by the system from the specified value due to factors such as the buffer size of the port and the buffer capability of the remote station. The final value will not exceed the specified value.

This option is useful when configuring to a host that has specific requirements for the I-field size but does not provide negotiation by way of XID exchanges.

TRACE Link?

This field specifies whether you want the system to save information about the activity on the link. A link trace is a sequential log of events that occur on the link that may be helpful in finding the source of a recurring error. However, performing a link trace uses processor and link time, as well as system storage. Valid values are the following:

`yes` Select this value to instruct the system to save link trace information about the link. Select this value only if you experience trouble with the link and need the information to help locate the problem.

Defining Token-Ring Physical Link Characteristics

The Token-Ring physical link profile defines characteristics of a Token-Ring link. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on a token ring define at least one Token-Ring physical link profile.

You can define more than one Token-Ring physical link profile, but only one set of conditions, or profile, can be active on a particular port. The attachment profile (see Defining Token-Ring Attachment Characteristics on page 13–135) designates the physical link profile that is active on a particular attachment.

The following procedure describes how to add a Token-Ring physical link profile, using the SMIT Interface

Entering Token-Ring Physical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `Token Ring.`
7. From the next SMIT menu, select `Token Ring Data Link Control.`
8. From the next SMIT menu, select `Token Ring Physical Link.`
9. From the next SMIT menu, select `Add a Profile.`
10. This displays the `Add SNA Token Ring Physical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Maximum number of LOGICAL LINKS

This field defines the maximum number of logical links that SNA can activate on this port at one time. The value for this field affects the local system's performance and response time for exchanges with remote systems, as well as its ability to establish a logical link with a remote site.

Lower values allow non-SNA logical links to share the same data link and adapter, but restrict the ability of SNA to establish an attachment. Higher values provide SNA with better access to the network, but restrict non-SNA attachments. The default value is 32.

Local SAP address

Enter the service access point (SAP) address for the transaction program on the local system. This address is a hexadecimal value in the indicated range. It does not need to be unique within the network. If you are setting up a new network, assign a value of `04` as the SAP address for each SNA Service in the network.

Changing the Default Characteristics

The following profile example shows the information fields that constitute the attachment profile. Once the new attachment profile appears on the screen, you must enter some additional information to correlate the attachment with the other profiles that describe it. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press the F7 (`Enter=Do`) key to save the new profile with the modified parameters.

Add SNA X.25 Attachment Profile Dialog

```
PROFILE name                                                    ...............
CONTROL POINT profile name                                      CDEFAULT
LOGICAL LINK profile name                                       QDEFAULT
PHYSICAL LINK profile name                                      QDEFAULT
STOP ATTACHMENT on inactivity?                                  no
  If yes, inactivity TIMEOUT (0–10 Minutes)                     0
LU address REGISTRATION?                                        no
  If yes, LU address REGISTRATION PROFILE name                  ...............
X.25 LEVEL                                                      1984
CALL Type                                                       listen
  If listen,
    AUTO-LISTEN?                                          no
    LISTEN NAME                                                 IBMQLLC
  If call, VIRTUAL CIRCUIT type                                 switched
    If permanent,
      Logical CHANNEL number of PVC (1–4095)                    1
    If switched,
      Remote station X.25 address                               ...............
      Optional X.25 facilities?                                 no
        If yes,
          REVERSE CHARGING?                                     no
          RPOA?                                                 no
            If yes, DATA NETWORK ID codes                       ...............
          PACKET size for RECEIVED data                         128
          PACKET size for TRANSMIT data                         128
          WINDOW size for RECEIVED data (1–127)                 2
          WINDOW size for TRANSMIT data (1–127)                 2
          THROUGHPUT Class for RECEIVED data                    9600
          THROUGHPUT Class for TRANSMIT data                    9600
          CLOSED USER group?                                    no
            If yes, INDEX to closed group                       ...............
          Closed user group with OUTGOING ACCESS?               no
            If yes, INDEX to closed group w/ OUT. ACCESS        ...............
          Network USER IDentification?                          no
```

In general, unless keeping the attachment open is very important, select yes and choose a value for the time-out period.

yes
: Select this value to stop the attachment after a period of inactivity. If you select this value, you must specify the length of the inactivity period in the following field.

no
: Select this value to remain attached to the remote node regardless of how long the connection remains idle.

Inactivity TIMEOUT

This field should only be changed and is required if the STOP CONNECTION on inactivity? field is yes. Enter a value in the specified range (0–10) for the number of minutes to wait before stopping the attachment. The actual waiting period is within 30 seconds of the specified number of minutes.

LU address REGISTRATION?

This field specifies whether LU addresses is registered for use by the attachment with the generic SNA application.

yes
: Select this value if you are using generic SNA and LU addresses are registered. Enter the name of the profile containing the list of registered LU addresses in the LU address REGISTRATION PROFILE name field.

no
: Select this value if you are not using generic SNA and do not need registered LU addresses.

LU address REGISTRATION PROFILE name

This field should only be changed if the LU address REGISTRATION? field is yes. This field provides the name of the LU address registration profile name that contains a list of LU addresses to be registered for use by this attachment with generic SNA.

X.25 LEVEL

This field refers to the date of the release of the CCITT X.25 recommendation that is supported. The two valid options are 1980 and 1984. Select the level required. Selecting 1984 enables the additional 1984 facilities and use of extended Cause Codes and Diagnostics.

CALL type

This field indicates whether the local station initiates a connection or receives requests for a connection from another station. Select one of the following values that describes how the local station operates:

call
: Select this value to indicate that the local station initiates a connection by calling another station.

listen
: Select this value to indicate that the local station does not initiate a connection but waits for a remote station to make a connection with it.

RPOA?

This field should only be changed if the `VIRTUAL CIRCUIT type` field is `switched`, the `CALL type` field is `call`, and the `Optional X.25 facilities?` field is `yes`. The `RPOA?` field indicates whether the Recognized Private Operating Agency network is accessed.

`yes`	Select this value if the RPOA network is accessed. If this value is selected, you may enter up to 30 4-digit codes separated by commas in the Data Network Identification Code field.
`no`	Select this value if the RPOA network is not accessed.

DATA NETWORK ID code

This field should only be changed if the `VIRTUAL CIRCUIT type` field is `switched`, the `Optional X.25 facilities` field is `yes`, the `CALL type` field is `call`, and the `RPOA?` field is `yes`. The `DATA NETWORK ID code` field identifies the RPOA transit network through an international gateway. Enter up to 30 four decimal digit codes separated by commas. At least one code must be entered. If more than one code is entered, single quotes have to surround the list, for example:

```
—A '3243,7574,9742,4821'
```

PACKET size for RECEIVED data

This field should only be changed if the `VIRTUAL CIRCUIT type` field is `switched`, the `CALL type` field is `call`, and the `Optional X.25 facilities` field is `yes`. The `PACKET size for RECEIVED data` field specifies the size (in octets) of the packets the calling DTE wants to receive from the called DTE. This value is negotiable.

Note: Note that with the 1980 level support, the maximum packet size is 1024. Check your support level configuration at the device level and in the `X.25 LEVEL` field.

PACKET size for TRANSMIT data

This field should only be changed if the `VIRTUAL CIRCUIT type` field is `switched`, the `CALL type` field is `call`, and the `Optional X.25 facilities` field is `yes`. The `PACKET size for TRANSMIT data` field specifies the size (in octets) of the packets the calling DTE wants to transmit to the called DTE. This value is negotiable

Note: Note that with the 1980 level support, the maximum packet size is 1024. Check your support level configuration at the device level and in the `X.25 LEVEL` field.

WINDOW size for RECEIVED data

This field should only be changed if the `VIRTUAL CIRCUIT type` field is `switched`, the `CALL type` field is `call`, and the `Optional X.25 facilities` field is `yes`. Select the requested window size for data received by the calling DTE. This value can range from 1 to 127. The default value for both directions is 2.

Note: Note that with the 1980 level support, the maximum window size is 7. Check your support level configuration at the device level and in the `X.25 LEVEL` field.

The Closed user group with OUTGOING ACCESS? field specifies whether the call is made within a closed user group with outgoing access. If the Closed user group with OUTGOING ACCESS? field is yes, the CLOSED USER group? field must be no.

yes — Select yes if the call is made within a closed user group with outgoing access. If you select yes, you may fill in the INDEX to closed user group field, however the CLOSED USER group? field must be no.

no — Select no if the call is not made within a closed user group with outgoing access.

INDEX to closed group with OUTGOING ACCESS

This field should only be changed if the CLOSED USER group? or the Closed user group with OUTGOING ACCESS? field is yes, the CALL type field is call, the Optional X.25 facilities? field is yes, and the VIRTUAL CIRCUIT type field is switched. Enter a four-digit number that specifies the closed user group within which the call is to be placed. If the number has less than four digits, insert leading zeros. The network provider allocates identifying codes for any closed user groups to which you subscribe.

Network USER IDentification?

This field should only be changed if the CLOSED USER group? field is yes, the CALL type field is call, the Optional X.25 facilities? field is yes, the VIRTUAL CIRCUIT type field is switched and if you have configured the X.25 device to support 1984 facilities and have selected 1984 support in the X.25 LEVEL field. The Network USER IDentification? field specifies whether the call is made with network user identification supplied in a facility block.

yes — Select yes if the call is made with network user identification supplied in a facility block.

no — Select no if the call is not made with network user identification supplied in a facility block.

Network USER ID name

This field should only be changed if the CLOSED USER group? field is yes, the CALL type field is call, the Optional X.25 facilities? field is yes, the VIRTUAL CIRCUIT type field is switched, the Network USER IDentification? field is yes, and if you have configured the X.25 device to support 1984 facilities and have selected 1984 support in the X.25 LEVEL field.

This field allows you to enter the Network User Identification information you want to supply the network when the call is placed.

Note: Note that the format of the information is defined by the Network Administration, and QLLC passes the data exactly in the form you supply it.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t log_x.25 ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the QLLC logical link profile. Once the new QLLC logical link profile appears on the screen, you can accept the default values for the parameters. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA QLLC Logical Link Profile Dialog

PROFILE name	....
DROP LINK on inactivity?	no
FORCE DISCONNECT timeout (1–600 seconds)	120
DEFINITION of maximum I-FIELD size	system_defined
If user-defined, max. I-FIELD SIZE (265–30729)	1417
TRACE Link?	no
If yes, TRACE SIZE	short
STATION type	secondary
If secondary or negotiable,	
Secondary INACTIVITY timeout	30
If primary or negotiable,	
Primary repoll TIMEOUT (1–255 seconds)	30
Primary repoll COUNT (1–255)	10

Figure 21. Add SNA QLLC Logical Link Profile Dialog

The following paragraphs supply information to help you choose the values that best describe your network.

PROFILE name

This field requests a name for the new logical link profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the attachment profile (see Defining X.25 Attachment Characteristics on page 13–149) for the attachments that use this logical link profile. Do not change this name without changing it throughout the entire profile.

`yes` Select this value to instruct the system to save link trace information about the link. Select this value only if you experience trouble with the link and need the information to help locate the problem.

`no` Select this value to instruct the system not to save link trace information. Select this value for normal operation of the link.

Refer to SMIT Problem Determination or the **trcrpt** command in *Commands Reference* for information on getting and formatting the link trace information once it has been saved. To access SMIT Problem Determination, type `smit problem` on the AIX command line, and press Enter. Refer to the **traceson** and **tracesoff** commands in *Commands Reference* for information on starting and stopping link traces from the operating system command line.

TRACE SIZE

This field should only be changed if the `TRACE Link?` field is `yes`. Use the following information to determine the correct selection for your system:

`short` Selects the *short* level of trace reporting, which saves information on approximately 77 of the latest link activities. Each entry in the log can be up to 80 bytes long, allowing room for approximately 48 bytes of send or receive data to be saved.

`long` Selects the *long* level of trace reporting, which saves information on approximately 24 of the latest link activities. Each entry in the log can be up to 256 bytes long, allowing room for approximately 224 bytes of send or receive data to be saved.

STATION type

This field defines the role of the local link station. A station may be primary, secondary, or of negotiable status.

`primary` Can issue command packets to a remote station and be responsible for managing the link.

`secondary` Responds to commands from a remote primary station. This is the default value.

`negotiable` Able to assume the role of either a primary or a secondary station, depending on the specific remote link station to which it is connected.

Secondary INACTIVITY timeout

This field should only be changed if the `STATION type` field is `secondary` or `negotiable`. The `Secondary INACTIVITY timeout` field specifies the time in seconds that a secondary station waits to receive a command packet from a primary station before starting inactivity procedures. Depending on the setting for the `DROP LINK on inactivity?` field, the link may be closed. The value may range from 1 to 255. The default value is 30.

Primary repoll TIMEOUT

This field should only be changed if the `STATION type` field is `primary` or `negotiable`. The `Primary repoll TIMEOUT` field specifies the time in seconds before a primary station retransmits an unacknowledged Q-packet. The value may range from 1 to 255. The default value is 30.

Defining X.25 Physical Link Characteristics

The X.25 physical link profile defines characteristics of an X.25 link. The system uses the information in this profile to determine how the network operates on the associated attachment. If you are using AIX SNA Services/6000 on an X.25 network, you must define at least one X.25 physical link profile.

You can define more than one X.25 physical link profile, but only one set of conditions, or profile, can be active on a particular port. The attachment profile (see Defining X.25 Attachment Characteristics on page 13–149) designates the physical link profile that is active on a particular attachment. Refer to the X.25 Communications Overview for System Management in *Communication Concepts and Procedures* for more information on the X.25 network.

The following procedure describes how to add an X.25 physical link profile, using the SMIT Interface

Entering X.25 Physical Link Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Enter `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Physical Units.`
6. From the next SMIT menu, select `X.25.`
7. From the next SMIT menu, select `X.25 Data Link Control.`
8. From the next SMIT menu, select `X.25 Physical Link.`
9. From the next SMIT menu, select `Add a Profile.`
10. This displays the `Add SNA X.25 Physical Link Profile` dialog. Add any names and change any default values necessary to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile`, `Remove a Profile or Alias`, `Print Profile(s)`, `Add an Alias for a Profile`, and `Change an Alias for a Profile`.

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Defining Generic LU Address Registration Characteristics

The Generic LU Address Registration profile contains a list of LU addresses that are registered for use by an attachment with the generic SNA application. In order for an attachment to use this list, the name of the appropriate LU address registration profile must be entered on the attachment profile. Refer to the following sections for more information on attachment profiles:

- Defining SDLC Attachment Characteristics on page 13–56
- Defining Standard Ethernet Attachment Characteristics on page 13–109
- Defining IEEE 802.3 Ethernet Attachment Characteristics on page 13–122
- Defining Token-Ring Attachment Characteristics on page 13–135
- Defining X.25 Attachment Characteristics on page 13–149.

The following procedure describes how to add a generic LU address registration profile, using the SMIT Interface

Entering Generic LU Address Registration Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Logical Units.`
6. From the next SMIT menu, select `Generic LU Address Registration.`
7. From the next SMIT menu, select `Add a Profile.`
8. This displays the `Add SNA LU Address Registration Profile` dialog. Add any names and change any default values to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

First, no profile name is displayed, when adding a profile, until the user enters one. When the user is changing a profile however, a name select dialog requests the name of the profile for which a change is desired. The change dialog is the same as the add dialog except that the current profile name is displayed and the `NEW PROFILE name` field allows the user to change the current profile name.

Second, the change dialog allows deletion of already registered addresses through the use of the `Unregister LU addresses` field.

Change SNA LU Address Registration Profile Dialog

CURRENT PROFILE name ....
NEW PROFILE name ....
Unregister LU addresses ....
Register LU addresses:
  Address 1 ....
  Address 2 ....
  Address 3 ....

  Address 31 ....

Figure 24. Change SNA LU Address Registration Profile Dialog

CURRENT PROFILE name

This field contains the name of the currently displayed LU address registration profile. The system uses this name to refer to the LU address registration profile associated with the profile and to refer to the set of characteristics described in that profile. The user cannot change the `PROFILE name` field in the change profiles.

NEW PROFILE name

This field is for changing the existing profile name. Enter the new profile name to be added. For this field, choose any name within the restrictions described in AIX SNA Services/6000 Naming Requirements on page 13–25 to refer to the LU address registration profile.

Unregister LU addresses

This field allows the user to unregister previously registered LU address profiles from the LU address registration profile. If you no longer need a profile or if a profile is no longer valid and you want to delete it, use the `Unregister LU addresses` field. You can press F4 (`F4=List`) for a list of currently registered LU addresses.

Select the LU address(es) to be unregistered, and press Enter (`Enter=Do`). Otherwise, enter the name of the profile to be deleted and press Enter. Valid values for the *Address* variable are 1 to 255. If you are deleting more than one profile, separate the profile names with a space.

Register LU addresses

This field contains a list of up to thirty-one LU addresses. Enter any new LU addresses as decimal numbers between 1 and 255. The LU addresses are used by an attachment with the generic SNA application.

Changing the Default Characteristics

The following example shows the information fields that constitute the local LU profile. Once the new local LU profile appears on the screen, you must enter some additional information to describe the particular application program. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press Enter (`Enter=Do`) to save the new profile with the modified parameters.

Add SNA LU1 Local LU Profile Dialog

PROFILE name	....
NETWORK name	....
Local LU name	....
Local LU address (1–255)	1
SSCP ID	000000000000

Figure 25. Add SNA LU1 Local LU Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests the name of the new local LU profile. The system uses this name to refer to the local LU associated with the profile and to refer to the set of characteristics that you describe in this profile. For this field, choose any name within the restrictions described in AIX SNA Services/6000 Naming Requirements on page 13–25 to refer to the local LU. When choosing a name, try to make the name describe the local LU. For example, a node located in the Publications department in Austin might be named:

```
Austin_Pubs
```

NETWORK name

This field provides the name of the network to which this LU is attached. The network name distinguishes the network from other networks to which it may be connected. This name, combined with the local LU name, forms a unique identifier (the *fully qualified name*) for the local LU within all networks to which it may be connected.

If you are attaching to an existing network, use the name of that network for this field. If you are creating a new network, choose any name within the restrictions described in AIX SNA Services/6000 Naming Requirements on page 13–25 to refer to the network.

Local LU name

This field provides the name of the local LU. The name entered here on the *local* system must match the name entered in the `REMOTE LU name` field of the connection profile on the *remote* system. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions associated with selecting a name for this field.

See Defining LU Type 1 Connection Characteristics on page 13–170 for a description of the connection profile.

Defining LU Type x (1,2,3) Connection Characteristics

The SNA connection profile contains fields that describe characteristics of the connection to a remote LU. AIX SNA Services/6000 uses the information in this profile to establish a connection with the remote LU and to associate other profiles with that connection. Each connection to a remote LU that AIX SNA Services/6000 establishes must have a connection profile defined for it.

The following procedure describes how to add an LU type *x* (1,2,3) logical connection profile, using the SMIT Interface

Entering LU Type x (1,2,3) Connection Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Logical Units.`
6. From the next SMIT menu, select `LU1, LU2,` or `LU3.`
7. From the next SMIT menu, select `LU1, LU2,` or `LU3 Logical Connection.`
8. From the next SMIT menu, select `Add a Profile.`
9. This displays the `Add SNA LU` *`x`* `(1,2,3) Logical Connection Profile` dialog. Add any names and change any default values to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t connection –u lu1 ProfileName

mksnaobj –t connection –u lu2 ProfileName

mksnaobj –t connection –u lu3 ProfileName
```

NETWORK name

This field provides the name of the network on which this connection is implemented. The network name distinguishes this network from other networks to which it may be connected. This name, combined with the remote LU name, forms a unique identifier (the *fully qualified name*) for the remote LU within all networks to which it may be connected. If you are creating a new network, choose a meaningful string of characters for the network within the restrictions defined below. If you are connecting to an existing network, enter the name of that network.

See Defining LU Type 1 Local Logical Unit Characteristics on page 13–167 or Defining LU Type 2 and LU Type 3 Local Logical Unit Characteristics on page 13–174 for more information on the remote LU name. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on selecting a name for this field.

REMOTE LU name

This field provides the name of the remote LU or adjacent CP name that forms the other half of this connection. The name entered here on the *local* system must match the name entered in the `Local LU name` field of the local LU profile on the *remote* system.

See Defining LU Type 1 Local Logical Unit Characteristics on page 13–167 or Defining LU Type 2 and LU Type 3 Local Logical Unit Characteristics on page 13–174 for a description of the local LU profile.

STOP CONNECTION on inactivity?

This field specifies whether AIX SNA Services/6000 should drop the connection if no application is using it (no outstanding opens) for a specified period of time. This choice affects system performance because each open connection uses system resources. If the connection is idle, then the resources may be used more efficiently elsewhere.

However, restarting the connection takes time to resolve the profiles associated with a connection, and you may want to keep the connection active to avoid the delay. In addition, if the link is not terminated when the stop occurs, cleanup procedures that require interaction with other nodes may not be completed. If this happens, the link must be put back into an active state for a normal stop to occur.

In general, unless keeping the connection open is very important, select `yes` and choose a value for the time-out period.

`yes` Select this value to drop the connection LU after a period of inactivity. If you select this value, the length of the inactivity period must be specified with the `TIMEOUT` field.

`no` Select this value to remain active regardless of how long the connection remains idle.

TIMEOUT

This field should only be changed and is required if the `STOP CONNECTION on inactivity?` field is `yes`. Enter a value in the specified range for the number of minutes to wait for a response before disconnecting the remote LU. The actual waiting period is within 30 seconds of the specified number of minutes.

Defining LU Type x (2,3) Local Logical Unit Characteristics

The local LU profile contains fields that describe characteristics of the local LU. AIX SNA Services/6000 uses the information in this profile to establish an attachment with the network and to associate a transaction program with that attachment. Each LU that is implemented on the local system must have a local LU profile defined for it.

The following procedure describes how to add an LU type *x* (2,3) local logical unit profile, using the SMIT Interface

Entering LU Type X (2,3) Local Logical Unit Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.

2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`

3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Logical Units.`
6. From the next SMIT menu, select `LU2` or `LU3.`
7. From the next SMIT menu, select `LU2` or `LU3 Local Logical Unit.`
8. From the next SMIT menu, select `Add a Profile.`
9. This displays the `Add SNA LU x (2,3) Local LU Profile` dialog. Add any names and change any default values to assure that the profile is accurate, and then press Enter (`Enter=Do`) to add the profile to the SNA database.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t local_lu –u lu2 ProfileName

mksnaobj –t local_lu –u lu3 ProfileName
```

Local LU address

This field defines the address that other systems use as the destination address field (DAF) to send information to the local LU. This address is defined by the host system to which this node is connected. Contact the network administrator at the host system to find out the value to enter in this field.

Enter the address of the local LU as a decimal number between 1 and 255.

SSCP ID

This field specifies the ID of the controlling system services control point (SSCP) in the SNA network. Enter the SSCP ID in hexadecimal for the LU. This identifier is defined by the host system to which this node is connected. Contact the network administrator at the host system to find out the value to enter in this field. This field is used for host verification. Entering an * (asterisk) allows any host to initiate sessions with this LU, as long as its `Local LU address` field matches that of a received session initiation request from a host.

Number of ROWS

This field defines the number of rows (horizontal lines of output) that the LU provides. Usually, this number refers to the number of lines that are available for output on the terminal screen that the application program is using. If this number is smaller than the number of lines of output that the host expects, the host rejects the BIND request. Enter a decimal number between 1 and 255.

Number of COLUMNS

This field defines the number of columns (vertical columns, or characters per line) that the LU provides. Usually, this number refers to the number of columns that are available for output on the terminal screen that the application program is using. If this number is smaller than the number of columns of output that the host expects, the host rejects the BIND request. Enter a decimal number between 1 and 255.

Changing the Default Characteristics

The following example shows the information fields that constitute the local LU profile. Once the new local LU profile appears on the screen, you must enter some additional information to describe the particular application program. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press Enter (`Enter=Do`) to save the new profile with the modified parameters.

Add SNA LU6.2 Local LU Profile Dialog

```
PROFILE name                                    ...............
TPN LIST profile name                           TDEFAULT
CP SESSION capable?                             no
  If no,
    NETWORK name                                ...............
    Local LU NAME                               ...............
    INDEPENDENT LU?                             yes
      If no,
        Local LU ADDRESS                        1
        SSCP ID                                 000000000000
```

Figure 28. Add SNA LU6.2 Local LU Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests the name of the new local LU profile. The system uses this name to refer to the local LU associated with the profile and to refer to the set of characteristics that you describe in this profile. For this field, choose any name within the restrictions described in AIX SNA Services/6000 Naming Requirements on page 13–25 to refer to the local LU. When choosing a name, try to make the name describe the local LU. For example, a node located in the Publications department in Austin might be named:

```
Austin_Pubs
```

TPN LIST name

This field specifies the name of the TPN list profile that lists the transaction program name profiles that can use this LU. You must define a TPN list profile with the name specified in the `TPN LIST name` field, and that profile must contain the name of at least one valid transaction program name profile.

See Defining LU 6.2 Transaction Programs for a Session on page 13–201 and Defining LU 6.2 Transaction Program Characteristics on page 13–195 for more information on TPN list profiles and transaction program name profiles.

SSCP ID

This field should only be changed if the INDEPENDENT LU? field is no. Enter the SSCP ID (System Services Control Point Identifier) in hexadecimal for the LU. This identifier is defined by the host system to which this node is connected. Contact the network administrator at the host system to find out the value to enter in this field. This field is used for host verification. Entering an * (asterisk) allows any host to initiate sessions with this LU, as long as its LOCAL LU address field matches that of a received session initiation request from a host.

Changing the Default Characteristics

The following example shows the information fields that constitute the connection profile. Once the new connection profile appears on the screen, you must enter some additional information to describe the particular application program. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press Enter (`Enter=Do`) to save the new profile with the modified parameters.

Add SNA LU6.2 Connection Profile Dialog

PROFILE name	....
ATTACHMENT profile name	RDEFAULT
LOCAL LU profile name	LDEFAULT
NETWORK name	....
STOP CONNECTION on inactivity?	no
If yes, TIMEOUT (0–10 Minutes)	0
CP SESSION capable?	no
If yes, CP NAME	....
If no,	
REMOTE LU name	....
REMOTE TPN LIST profile name	RDEFAULT
MODE LIST profile name	MDEFAULT
INTERFACE type	extended
If extended, SESSION CONCURRENCY	single
Node VERIFICATION?	no

Figure 29. Add SNA LU6.2 Connection Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests the name of the new connection profile. The system uses this name to refer to the remote LU associated with the profile and to refer to the set of characteristics that you describe in this profile. For this name, you may choose to use the name of the remote system to which you are connected. The connection profile name is the name that you use to start or stop a connection, using the **startsrc** and **stopsrc** commands.

ATTACHMENT profile name

This field provides the name of the attachment profile that describes the characteristics of the attachment to the remote LU. You must create an attachment profile having the name that you provide in this field. Refer to one of the following sections for information about creating this profile:

- Defining SDLC Attachment Characteristics on page 13–56
- Defining Standard Ethernet Attachment Characteristics on page 13–109
- Defining IEEE 802.3 Ethernet Attachment Characteristics on page 13–122
- Defining Token-Ring Attachment Characteristics on page 13–135
- Defining X.25 Attachment Characteristics on page 13–149.

CP NAME

This field should only be changed if the CP SESSION capable? field is yes. The CP NAME field provides the name of the remote CP name that forms the other half of this connection. The name entered here on the *local* system must match the name entered in the LOCAL LU name field of the local LU profile on the *remote* system.

See Defining LU Type 6.2 Local Logical Unit Characteristics on page 13–177 for a description of the local LU profile.

REMOTE LU name

This field should only be changed if the CP SESSION capable? field is no. The REMOTE LU Name field provides the name of the remote LU or adjacent CP name that forms the other half of this connection. The name entered here on the *local* system must match the name entered in the LOCAL LU name field of the local LU profile on the *remote* system. See Defining LU Type 6.2 Local Logical Unit Characteristics on page 13–177 for a description of the local LU profile.

REMOTE TPN LIST profile name

This field should only be changed if the CP SESSION capable? field is no. The REMOTE TPN LIST profile name field specifies the name of the remote transaction program name (RTPN) list profile that lists the remote TPN profiles applying to this connection. You must define a remote TPN list profile with the name specified here, and that profile must contain the name of at least one valid remote TPN profile.

The remote TPN list, mode list, and TPN list allow the SNA program to resolve multiple profiles at once (providing better performance for the application). They also provide a certain amount of control by the administrator for multiple applications running simultaneously. All the transaction programs listed run on the same connection.

See Defining LU Type 6.2 Remote Transaction Programs for a Session on page 13–208 and Defining LU Type 6.2 Remote Transaction Program Characteristics on page 13–204 for more information on remote TPN list profiles and remote TPN profiles.

MODE LIST profile name

This field should only be changed if the CP SESSION capable? field is no. This field specifies the name of the mode list profile that lists the mode profiles applying to this connection. You must define a mode list profile having the name specified here, and that profile must contain the name of at least one valid mode profile.

See Defining LU Type 6.2 Modes for a Session on page 13–191 and Defining LU Type 6.2 Session Characteristics on page 13–187 for more information on mode list profiles and mode profiles.

INTERFACE type

This field should only be changed if the CP SESSION capable? field is no. This field specifies to AIX SNA Services/6000 the type of subroutine interface that the program uses. Contact the programmer responsible for the program to get this information. If you cannot find out, select extended.

If you select yes and then want to change or remove a BIND password, or if you select no and then change your mind and want to add a BIND password, use the following procedure to add, change, or remove a BIND password, using the SMIT Interface.

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Security.`
6. From the next SMIT menu, select `Add a BIND Password.`
7. This displays a name select dialog requesting the `LU type 6.2 CONNECTION profile name`. If you don't know the connection profile name, press F4 (F4=List) to display a list of all known connections. Select the connection for which the BIND password is desired, and press Enter (Enter=Do).
8. This displays a pop-up panel requesting the password and whether the password is in hexadecimal. Enter the new password in accordance with the instructions.
9. You must reenter the password to confirm that it has been entered correctly. Reenter the new password, and press Enter to add the password.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a BIND password. From the same menu that you selected `Add a BIND Password`, you can also select `Change a BIND Password`, or `Remove a BIND Password`. The last procedural step for both changing and removing a BIND password is nearly identical to that for adding a BIND password. Refer to Adding the BIND Password on page 13–235, Changing the BIND Password on page 13–236, or Removing the BIND Password on page 13–237 for more procedural information.

If you want node verification on a connection between an AIX node and another LU 6.2 system, such as a System 36, the other LU 6.2 system must have a security mode defined for the secure connection. The attributes of the security mode for the other system are:

Mode Name	SNASVCRB
Maximum Number Sessions	2
Minimum First Speakers	1
Minimum Bidders	1

Do not add this mode to your AIX node profiles. AIX SNA Services/6000 includes this information.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj –t mode ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the mode profile. Once the mode profile appears on the screen, you can choose to accept the default values for the parameters. To save the new profile with only the default parameters, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified parameters.

Add SNA LU6.2 Mode Profile Dialog

PROFILE name	....
MODE name	....
Maximum number of SESSIONS (1–999)	1
Minimum contention WINNERS (0–499)	0
Minimum contention LOSERS (0–500)	0
RECEIVE pacing (0–63)	3
SEND pacing (0–63)	3
Maximum RU SIZE (256, 288, ..., 3840)	2816
RECOVERY level	no_reconnect

Figure 30. Add SNA LU6.2 Mode Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

PROFILE name

This field requests a name for the new mode profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the mode list profile (see Defining LU Type 6.2 Modes for a Session on page 13–191) for the sessions that use this mode profile to define their session characteristics. Do not change this name without changing it throughout the entire profile.

MODE name

This field identifies the set of rules and protocols to be used for the session when the local LU is connected as a dependent LU (to a host system). For peer-to-peer sessions (independent LUs), the primary LU sends the mode name in the BIND request at the start of the session.

You can select any valid string of characters (described in AIX SNA Services/6000 Naming Requirements on page 13–25) for the mode name. However, both LUs (local and remote) must use the same mode name for the session between them.

SEND pacing

This field, along with the `RECEIVE pacing` field, controls session-level pacing for the session. Session-level pacing helps control the rate that the Transmission Control portion of the LU processes requests during normal flow transmissions. It does not affect expedited flow transmissions. Use pacing when the sending LU can send requests faster than the receiving LU can process them.

The `SEND pacing` field controls the flow of outbound data. If the local LU can send requests faster than the remote LU can process them, enter a positive integer in the specified range in the `SEND pacing` field. This number specifies the number of request units (RUs) that the local LU can send after which it must wait for a pacing response from the remote LU. The best value varies with each installation. If you do not have an established value, start with the default value and adjust it, if needed, to avoid overflow. Enter a value of 0 to disable send pacing.

Maximum RU SIZE

This field determines the maximum number of bytes in each request/response unit (RU) used in this session and also determines the size of the buffer for sending and receiving data. This number, together with the send and receive pacing values, helps prevent an LU from receiving more data than it can process at one time. The LU uses this number during BIND request negotiation to establish the RU size for the session.

The value entered for both the local and remote LUs involved in the session should be the same. The best value varies with each installation. If you do not know a value, start with the default value and adjust it, if needed, tc avoid overflow.

To specify an RU size, enter one of the following valid positive integers:

256	288	320	352	384	416	448
480	512	576	640	704	768	832
896	960	1024	1152	1280	1408	1536
1664	1792	1920	2048	2304	2560	2816
3072	3328	3584	3840			

RECOVERY level

This field specifies whether the transaction programs can reconnect a conversation.

`reconnect` Select this option if the transaction programs can reconnect a conversation. To reconnect, the local and remote transaction programs must agree between themselves which program starts the reconnection. That program must then maintain the request ID of the conversation to be able to request the reconnection.

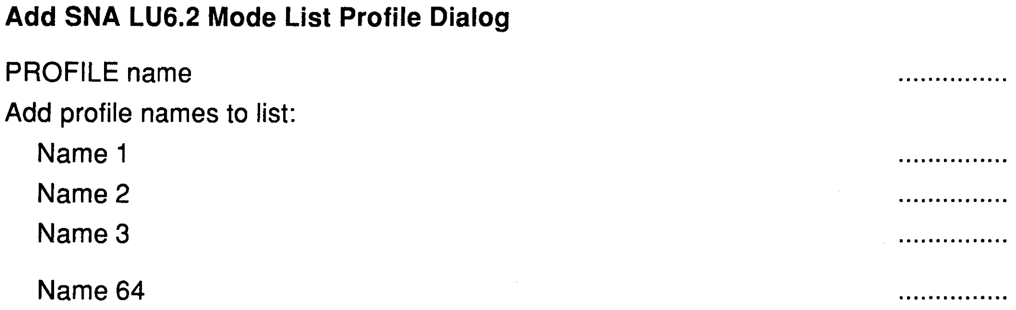
`no_reconnect` Specify this option if the transaction programs cannot reconnect a conversation. Selecting this option prevents the transaction programs from reconnecting.

Changing the Default Entries

The following profile example shows the information fields that constitute the mode list profile. Once the mode list profile appears on the screen, you can choose to accept the mode profiles listed in the default copy of the profile, or you can add, change or delete entries to suit your needs. When defining the mode list for a session, ensure that:

- Each application program that runs on the session has its mode profile listed in the mode list profile.
- You define each mode profile before entering them in the mode list profile (see Defining LU Type 6.2 Session Characteristics on page 13–187).

To add, change or delete an entry move the cursor to the affected field and then edit the field as needed to enter the proper mode profile names. To enter more profiles than space allows, fill in the fields that appear in the window and then scroll down to display additional blank fields. The mode list profile contains two field types as seen in the profile example below.



Add SNA LU6.2 Mode List Profile Dialog

PROFILE name ....

Add profile names to list:

Name 1 ....
Name 2 ....
Name 3 ....

Name 64 ....

Figure 31. Add SNA LU6.2 Mode List Profile Dialog

PROFILE name

This field requests a name for the new mode list profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

Add profile names to list

The remaining fields in the profile contain the names of mode profiles that can be selected for use on the session associated with this mode list. Each mode profile in this list must be a defined mode profile. See AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on allowed characters for a mode profile name.

The first four mode entries in the mode list can be selected by the application program when it runs by using the **writex** subroutine. This call uses the priority field of the **ext_io_str** structure to select one of these mode entries.

Add profile names to list

The remaining fields in the profile contain the names of mode profiles that can be selected for use on the session associated with this mode list. Each mode profile in this list must be a defined mode profile. See AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on allowed characters for a mode profile name.

The first four mode entries in the mode list can be selected by the application program when it runs by using the **writex** subroutine. This call uses the priority field of the **ext_io_str** structure to select one of these mode entries.

Generated Base Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated base command for this dialog is:

```
mksnaobj -t transact ProfileName
```

Changing the Default Characteristics

The following profile example shows the information fields that constitute the TPN profile. On the screen, you must scroll down to see the information at the bottom of the profile. Once the new TPN profile appears on the screen, you must enter some additional information to describe the particular application program. You cannot use only the default values provided.

To change or add to any of the values for the fields on the screen, move the cursor to the supplied value and enter the new value. When all values have been entered, press Enter (`Enter=Do`) to save the new profile with the modified parameters.

Add SNA LU6.2 TPN Profile Dialog

PROFILE name	....
Transaction program name is in HEXADECIMAL?	no
TRANSACTION program name	tpn
PIP data?	no
If yes, SUBFIELDS (0–99)	0
CONVERSATION Type	mapped
RECOVERY level	no_reconnect
SYNC level	none
Directory PATH	/bin/svr
MULTIPLE INSTANCES supported?	no
User ID	100
SERVER synonym name	....
RESTART action	once
COMMUNICATION type	signals
If IPC, communication IPC queue key	....
Standard INPUT file/device	/dev/console
Standard OUTPUT file/device	/dev/console
Standard ERROR file/device	/dev/console

Figure 33. Add SNA LU6.2 TPN Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests a name for the new TPN profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

The profile name also appears in the SNA TPN list profile (see Defining LU Type 6.2 Transaction Programs for a Session on page 13–201). Do not change this name without changing it throughout the entire profile.

RECOVERY level

This field specifies whether the remote transaction program can reconnect a conversation. Valid values for this field are:

`reconnect` Select this option if the remote transaction program can reconnect a conversation. To reconnect, the local and remote transaction programs must agree between themselves which program starts the reconnection. That program must then maintain the request ID of the conversation to be able to request the reconnection.

Selecting this option does not require the remote transaction program to reconnect, but it does provide the option of reconnecting.

`no_reconnect`

Specify this option if the remote transaction program cannot reconnect a conversation. Selecting this option prevents the remote transaction program from reconnecting.

SYNC level

This field specifies whether the transaction program can respond to or send CONFIRM requests from or to the local transaction program. A CONFIRM request from the local transaction program results in a positive or negative reply from the transaction program to indicate whether the transmitted data was successfully received.

`confirm` Select this option if the transaction program sends and responds to CONFIRM requests.

`none` Select this option if the transaction program does not send or respond to CONFIRM requests.

`either` Select this option if the transaction program is capable of sending and responding depending on the sync level support of the remote transaction program. Select this option also if you do not know whether the transaction program can send or respond to such a request.

Directory PATH

This field provides the full path name on the local system for the program. Since these programs, by convention, are usually stored in the /**bin** directory, that part of the path name is included in the default, which is /**bin**/**svr**. You must complete the path name with the name of the application program or change the path name if the application program is in a different directory.

For example, if the name of the application program is **svrprog** and it is in the /**bin** directory, enter the following path name for this field:

```
/bin/svrprog
```

MULTIPLE INSTANCES supported?

This field specifies whether multiple instances of this transaction program may run concurrently. Valid values for this field are:

`yes` Select this value if multiple instances of this transaction program may run concurrently.

`no` Select this value if only one instance of this transaction program may run at a time.

Communication IPC queue key

This field should only be changed if the COMMUNICATION type field is ipc. Enter a decimal value that corresponds to the IPC queue key that the program uses. This value is written into the program when the program is created. Refer to the documentation for that program to find out this value. If you do not know the value, select signals in the COMMUNICATION type field.

Standard INPUT file/device

This field specifies the file or device from which the transaction program receives its input while running. The default is **/dev/console**. For programs that run in the background without additional input, change the value to **/dev/null**. Otherwise, enter the name of the special file that corresponds to the input device for the program, or the name of a file that provides input values to the program.

Standard OUTPUT file/device

This field specifies the file or device to which the transaction program sends its output while running. The default is **/dev/console**. For programs that run in the background, change the value to **/dev/null** to prevent the transaction program from writing to the screen while another program is running. Otherwise, enter either the name of the special file that corresponds to the output device for the program or the name of a file to which the program can write its output.

Standard ERROR file/device

This field specifies the file or device to which the transaction program writes its error messages while running. The default is **/dev/console**. For programs that run in the background for which you do not want to save error messages, change the value to **/dev/null**. Otherwise, enter either the name of the special file that corresponds to the error output device for the program or the name of a file to which the program can write its error messages.

Changing the Default Entries

The following profile example shows the information fields that constitute the SNA TPN list profile. Once the SNA TPN list profile appears on the screen, you can choose to accept the profile names listed in the default copy of the profile, or you can add, change or delete entries to suit your needs. When defining the SNA TPN list for a session, ensure that:

- Each application program that runs on the session has its transaction program name profile listed in the SNA TPN list profile.
- You define each transaction program name profile before entering them in the SNA TPN list profile (see Defining LU Type 6.2 Transaction Program Characteristics on page 13–195).

To add, change or delete an entry move the cursor to the affected field and then edit the field as needed to enter the proper profile names. To enter more profiles than space allows, fill in the fields that appear in the window and then scroll down to display additional blank fields. The TPN list profile contains two field types as seen in the profile example below.

Add SNA LU6.2 Transaction Program Name List Profile Dialog

PROFILE name ....

Add profile names to list:

Name 1 ....

Name 2 ....

Name 3 ....

Name 64 ....

Figure 34. Add SNA LU6.2 Transaction Program Name List Profile Dialog

PROFILE name

This field requests a name for the new transaction program name (TPN) list profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions placed on choosing a name for this field.

Add profile names to list

The remaining fields in the profile contain the names of transaction program profiles that can be selected for use on the session associated with this transaction program name list. Each TPN profile in this list must be a defined TPN profile. See AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on allowed characters for a TPN profile name.

Change Profile Dialog

If the user elects to change a TPN list profile, the dialog displayed is similar to the dialog that is displayed when the user elects to add a TPN profile, with two differences.

First, no profile name is displayed, when adding a profile, until the user enters one. When the user is changing a profile however, a name select dialog requests the name of the profile for which a change is desired. The change dialog is the same as the add dialog except that the current profile name is displayed and the `NEW PROFILE name` field allows the user to change the current profile name.

Second, the change dialog allows deletion of already listed TPN profiles through the use of the `DELETE profile names from list` field.

Defining LU Type 6.2 Remote Transaction Program Characteristics

Note: Use this profile for LU 6.2 connections only.

The remote transaction program name (TPN) profile contains fields that describe characteristics of an application program on the remote LU. The local LU uses the information in this profile to request the start of the associated application program on the remote system, and to define the environment in which the conversation with that program occurs.

Each remote transaction program used with AIX SNA Services/6000 must have a remote TPN profile. To create this profile you may need to get information about the remote transaction program from the remote site. If the remote transaction program is the same as one of your local transaction programs, look in the TPN profile for that local program to determine the values to enter in the remote transaction name profile for the remote program. The remote TPN profile does not contain all of the information contained in the local TPN profile.

In addition to defining a remote TPN profile for each remote program, you must also list the name of each remote TPN profile in the SNA remote transaction program name (RTPN) list profile (see Defining Remote Transaction Programs for a Session on page 13–208). The LU can then select one of the transaction programs in the RTPN list profile with which to establish a conversation.

The following procedure describes how to add an LU type 6.2 RTPN profile, using the SMIT Interface

Entering LU Type 6.2 Remote TPN Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Logical Units.`
6. From the next SMIT menu, select `LU6.2.`
7. From the next SMIT menu, select
   `LU6.2 Remote Transaction Program Name (RTPN).`
8. From the next SMIT menu, select `Add a Profile.`
9. This displays the `Add SNA LU6.2 RTPN Profile` dialog. Add any names and change any default values to assure that the profile is accurate, and then press Enter (`Enter=Do`).

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

RTPN name is in HEXADECIMAL?

This field specifies whether the TPN is specified in hexadecimal or as a character string. Service transaction programs are specified in hexadecimal and should not be translated into EBCDIC before being transmitted to the remote LU. Character string names are stored and transmitted in EBCDIC. Refer to EBCDIC to ASCII Translation for US English (TEXT) on page 3–70 for assistance in converting ASCII to EBCDIC.

`yes` Select this option to indicate a hexadecimal TPN.

`no` Select this option to indicate a character string TPN.

RTPN name

This field specifies the name of the transaction program in the form selected (either hexadecimal or character string). Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on follow when creating either of these types of names.

PIP data?

This field specifies whether the remote transaction program expects to receive PIP (program initialization parameter) data when it starts up. This data may include option-selection information, additional file names, or other information that the remote transaction program uses to establish its operating environment. This data is usually in the form of a character string (or strings) that is passed to the remote transaction program when it is activated. PIP data is available to a C language program through the `argv[ ]` parameters.

`yes` Select this option if the remote transaction program requires PIP data.

`no` Select this option if the remote transaction program does not require PIP data or if you do not know whether it does.

CONVERSATION Type

This field specifies whether the transaction program operates on a basic or a mapped conversation. Only LU 6.2 sessions allow mapped conversations. They are used chiefly by application transaction programs. Service transaction programs use basic conversations. Refer to the documentation for the particular transaction program to determine the type of conversation it uses.

`basic` Select this option to indicate a basic conversation.

`mapped` Select this option to indicate a mapped conversation.

`either` Select this option to indicate that the transaction program may operate on either a mapped or basic conversation.

You can either select one conversation type or select both types concurrently.

Defining LU Type 6.2 Remote Transaction Programs for a Session

Note: Use this profile for LU 6.2 connections only.

The SNA remote TPN list profile provides a list of remote transaction program name (TPN) profiles to be available for use with a particular session. Each name in the list must be the name of a valid remote TPN profile (see Defining LU Type 6.2 Remote Transaction Program Characteristics on page 13–204). The local LU can choose any of the remote TPN profiles listed in the SNA remote TPN list profile to define the characteristics of a remote transaction program to start during a particular session.

The following procedure describes how to add an LU type 6.2 RTPN list profile, using the SMIT Interface

Entering LU Type 6.2 RTPN List Information

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Logical Units.`
6. From the next SMIT menu, select `LU6.2.`
7. From the next SMIT menu, select `LU6.2 Remote Transaction Program Name List.`
8. From the next SMIT menu, select `Add a Profile` or `Change a Profile.`
9. This displays the `Add SNA LU6.2 RTPN List Profile` or `Change SNA LU6.2 RTPN List Profile` dialog. Add any names and change any default values to assure that the profiles are accurate, and then press Enter (`Enter=Do`).

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

The preceding procedural steps explain how to add a profile to the SNA profile database. However, from the same step that users select `Add a Profile`, they can also select `Change a Profile, Remove a Profile or Alias, Print Profile(s), Add an Alias for a Profile,` and `Change an Alias for a Profile.`

Selecting `Change a Profile` displays a profile dialog that is identical to the add profile with two exceptions. The `PROFILE name` field contains the current profile name and cannot be changed, and an additional field, `NEW PROFILE name`, is supplied for users to change profile names. Selecting `Remove a Profile or Alias` displays a name select dialog requesting the profile name(s) and/or alias name(s) to be removed.

Change Profile Dialog

If the user elects to change an RTPN list profile, the dialog displayed is similar to the dialog that is displayed when the user elects to add an RTPN profile, with two differences.

First, no profile name is displayed, when adding a profile, until the user enters one. When the user is changing a profile however, a name select dialog requests the name of the profile for which a change is desired. The change dialog is the same as the add dialog except that the current profile name is displayed and the `NEW PROFILE name` field allows the user to change the current profile name.

Second, the change dialog allows deletion of already listed RTPN profiles through the use of the `DELETE profile names from list` field.

Change SNA LU6.2 RTPN List Profile Dialog

CURRENT PROFILE name ....

NEW PROFILE name ....

DELETE profile names from list ....

Add profile names to list:

Name 1 ....

Name 2 ....

Name 3 ....

Name 63 ....

Figure 38. Change SNA LU6.2 RTPN List Profile Dialog

CURRENT PROFILE name

This field contains the name of the currently displayed RTPN list profile. The system uses this name to refer to the RTPN list associated with the profile and to refer to the set of characteristics described in that profile. The user cannot change the `PROFILE name` field in the change profiles.

NEW PROFILE name

This field is for changing a profile name. Enter the new profile name to be added. For this field, choose any name within the restrictions described in AIX SNA Services/6000 Naming Requirements on page 13–25 to refer to the RTPN list profile. When choosing a name, try to make the name describes the RTPN list.

DELETE profile names from list

This field allows the user to delete previously listed RTPN profiles from the RTPN list profile. If you no longer need a transaction program profile or if a transaction program profile is no longer valid and you want to delete it, use the `DELETE profile names from list` field.

You can press F4 (`F4=List`) for a list of currently listed RTPN profiles. Select the RTPN profile(s) to be deleted, and press Enter (`Enter=Do`). Otherwise, enter the name of the profile to be deleted. If you are deleting more than one profile, separate the profile names with a space.

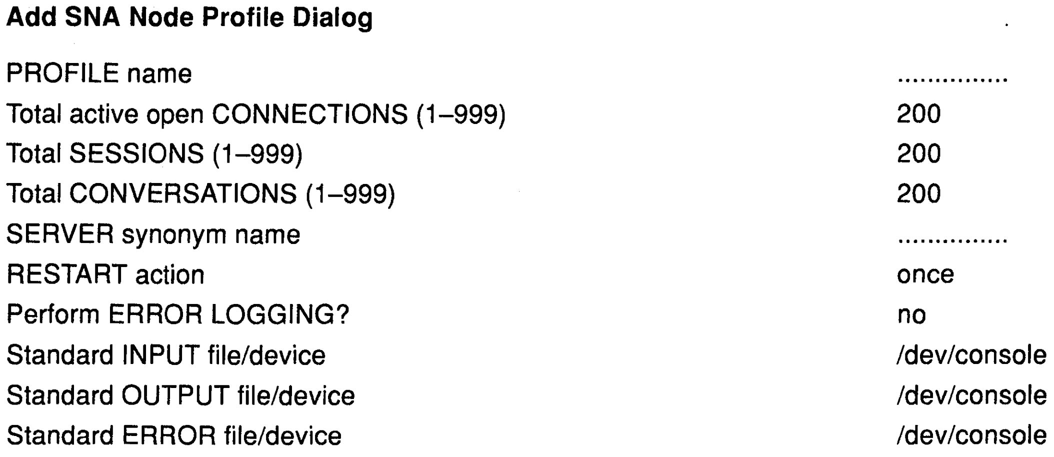
Add profile names to list

The remaining fields in the profile contain the names of remote TPN profiles that can be selected for use on the session associated with this list. Each profile in this list must be a defined remote TPN profile. See AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions on allowed characters for an RTPN profile name.

Changing the Default Characteristics

The following profile example shows the information fields that make up the SNA profile. Once the SNA profile appears on the screen, you can choose to accept the default values for the fields. To save the new profile with only the default fields, press Enter (`Enter=Do`).

You can also change any of the values for the fields on the screen by moving the cursor to the supplied value and entering the new value. When all values have been entered, press Enter to save the new profile with the modified fields.



Add SNA Node Profile Dialog

PROFILE name	....
Total active open CONNECTIONS (1–999)	200
Total SESSIONS (1–999)	200
Total CONVERSATIONS (1–999)	200
SERVER synonym name	....
RESTART action	once
Perform ERROR LOGGING?	no
Standard INPUT file/device	/dev/console
Standard OUTPUT file/device	/dev/console
Standard ERROR file/device	/dev/console

Figure 39. Add SNA Node Profile Dialog

The following paragraphs supply information to help choose the values that best describe your network.

PROFILE name

This field requests the name of the new SNA profile. The system uses this name to refer to the set of characteristics that you describe in this profile. You must always have a profile named `sna` that describes the characteristics of the operating environment for AIX SNA Services/6000. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions associated with selecting a name for this field.

Total active open CONNECTIONS

This field defines the maximum number of LU-LU pairs that can be active at any time. Choose the value for this field based on the number of open connections that must be active at the same time. The minimum value for this field should be at least twice the number of LUs that are active at any time.

Total SESSIONS

This field defines the maximum number of sessions that can be active at any time. The value for this field affects local system performance and response time for exchanges with remote systems, as well as the ability to start a session with a remote LU as required. Lower values provide better performance and response time, but restrict the ability to start a session. Higher values provide better access to the network, but you may experience lower system performance and longer response times.

Choosing a value is a trial-and-test process. Start with the default value (or some other value that you may choose within the specified range) and try it for a while. Then adjust the value up or down to get the desired balance of performance, response time, and access to the network. The minimum value for this field should be at least twice the number of LUs that are active at any time.

Standard OUTPUT file/device

This field specifies the file or device to which AIX SNA Services/6000 sends its output while it is running. The default is /**dev/console**.

Standard ERROR file/device

This field specifies the file or device to which AIX SNA Services/6000 writes its error messages while running. The default is /**dev/console**.

Add SNA Control Point Profile Dialog

PROFILE name	....
XID node ID	05c00000
NETWORK name	....
CONTROL POINT name	....
CP status	no_bind
LOCATE GDS supported?	no
DIRECTORY SERVICES supported?	no
RESOURCE registration supported?	no
Registration of CHARACTERISTICS supported?	no
REQUEST/REPLY CP-MSUs supported?	no
UNSOLICITED CP-MSUs supported?	no
PARALLEL CP-CP sessions supported?	no
TOPOLOGY database update supported?	no
If YES, flow reduction SEQUENCE NUMBER	0

Figure 40. Add SNA Control Point Profile Dialog

The following paragraphs supply information to help choose the values that best describe the application program.

PROFILE name

This field requests the name of the new profile. The system uses this name to refer to the set of characteristics that you describe in this profile. Refer to AIX SNA Services/6000 Naming Requirements on page 13–25 for the restrictions for choosing a name for the new profile.

XID node ID

This field provides the node ID of the remote physical unit. This value is the hexadecimal ID that is exchanged with the remote physical unit when a connection is first established (XID operation). The XID operation helps ensure that the correct physical units are being connected.

Note: A value of all zeroes or all ones for this field indicates that no unique identifier has been assigned for the XID node ID.

This 8-position field is actually two fields concatenated together. These fields are:

Block Number The first three hexadecimal digits of this field provide an identifier, or *block number*, that is unique to each different product on the network. For example, if the system is a RISC System/6000, use the characters `071` for the block number. Refer to the documentation for the specific product to find out its block number.

ID Number The last five hexadecimal digits in this field distinguish a specific piece of equipment from all other similar pieces of equipment on the network. The way to choose this number varies with the types of products that make up the network. If you are creating your own network of only RISC System/6000s, you can choose any unique identifier, such as a sequence number, for each member of the network. If you are attaching to a network that uses other equipment, refer to the documentation for that equipment for the recommended method for choosing an ID number.

RESOURCE registration supported?

This field indicates whether the local CP accepts REGISTER requests. The value returned is either yes or no.

yes — Specifies that the local CP does accept REGISTER requests.

no — Specifies that the local CP does not accept REGISTER requests.

Registration of CHARACTERISTICS supported?

This field indicates whether the local CP accepts REGISTER requests that include resource characteristics. The value returned is either yes or no.

yes — Specifies that the local CP does accept REGISTER requests that include resource characteristics.

no — Specifies that the local CP does not accept REGISTER requests that include resource characteristics.

REQUEST/REPLY CP-MSUs supported?

This field indicates whether the local CP accepts requests for management services data in a CP-MSU and replies to requests in a CP-MSU. The value returned is either yes or no.

yes — Specifies that the local CP does accept requests for management services data in a CP-MSU and replies to requests in a CP-MSU.

no — Specifies that the local CP does not accept requests for management services data in a CP-MSU and therefore does not reply to requests in a CP-MSU.

UNSOLICITED CP-MSUs supported?

This field indicates whether the local CP accepts unsolicited requests for management services data in a CP-MSU. The value returned is either yes or no.

yes — Specifies that the local CP does accept unsolicited requests for management services data in a CP-MSU.

no — Specifies that the local CP does not accept unsolicited requests for management services data in a CP-MSU.

PARALLEL CP-CP sessions supported?

This field indicates whether the local CP supports and activates parallel CP-CP sessions. The value returned is either yes or no.

yes — Specifies that the local CP does support and activate parallel CP-CP sessions.

no — Specifies that the local CP does not support and activate parallel CP-CP sessions.

Starting and Stopping AIX SNA Services/6000

Starting and Stopping AIX SNA Services/6000 explains how to start and stop AIX SNA Services/6000 manually by using the System Management Interface Tool (SMIT). For more information on the use of the System Resource Controller (SRC) commands for performing these operations, refer to the SRC Overview. Throughout this document, the SRC command is given for each SMIT function performed (for example, Start SNA, Start an SNA Attachment, Stop an SNA Connection, and so on).

Background Information

AIX SNA Services/6000 controls and monitors Logical Unit (LU) session establishment, manages mapping tables and routing for LU sessions, and provides Control Program (CP) services in order to establish an attachment to the Link Control (LC). AIX SNA Services/6000 and its resources can be in one of the following states:

active The resource can perform the functions for which it was designed and may or may not be in use.

inactive The device is not attached or not available for attachment to another device.

pending The server has received a request for an action (start or stop) but has not yet performed that action.

You can perform the operations in Starting and Stopping AIX SNA Services/6000 directly from the AIX command line, using System Resource Controller (SRC) commands, or indirectly, using the SMIT menus. SNA is a subsystem, controlled by the SRC. The **startsrc** and **stopsrc** commands provide the interface for starting and stopping SNA. For information on the SMIT Interface, refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures.* For more information about the System Resource Controller, refer to the SRC Overview in *General Concepts and Procedures.*

Generated Command

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this procedure is:

```
startsrc –ssna
```

Starting a Connection

A connection is the linking together of two or more logical units (LUs) on an attachment. The connection provides communications channels between the LUs for the application programs running at the respective LUs.

The following conditions must be satisfied before a connection can be started:

- Two systems must be set up to communicate with each other, since a connection cannot be started with one system only.
- The AIX SNA Services/6000 environment must be active before a connection can be started.
- If the associated attachment is not already active, AIX SNA Services/6000 starts it.
- A listening station waits on the line until a calling station starts the connection.
- The **startsrc** command must be issued on the listening station before it is issued on the calling station.

The following procedure describes how to start a connection, using the SMIT Interface.

Starting a Connection

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Control SNA Services`.
5. From the next SMIT menu, select `Start SNA Services`.
6. From the next SMIT menu, select `Start an SNA Connection`.
7. This displays a name select dialog asking for the `Connection Profile Name`. If you do not know the connection profile name, press the F4 key (F4=List) to display a list of all known connections. Select the connection to be started, and press Enter (`Enter=Do`) to start the connection.

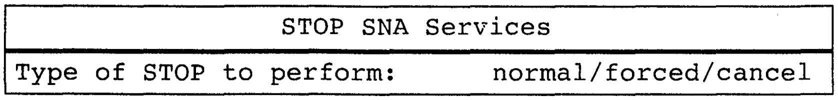
Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
startsrc –t connection –o ConnName
```

Stop AIX SNA Services/6000 Dialog



```
                    STOP SNA Services
Type of STOP to perform:      normal/forced/cancel
```

Figure 1. Stop AIX SNA Services/6000

Generated Commands

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated commands for this dialog are:

normal:	`stopsrc –ssna`
forced:	`stopsrc –f –ssna`
cancel:	`stopsrc –c –ssna`

Generated Commands

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

normal: `stopsrc –t connection –o` *ConnName*
forced: `stopsrc –f –t connection –o` *ConnName*

Generated Commands

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
stopsrc –t attachment –o AttName

stopsrc –f –t attachment –o AttName
```

Adding a Communication Authority Password

The following procedure explains how to add a communication authority password, using the System Management Interface Tool (SMIT).

Warning: Always back up your profile database before adding the communication authority password so that you can restore the system if you forget the password.

The user must define a new password when the node security type is changed from currently unsecured to currently secured. Once the node security type has been changed to currently secured, SNA Services prompts for the Communication Authority Password whenever an SNA command is issued.

The following procedure describes how to add a communication authority password, using the SMIT Interface.

Adding a Communication Authority Password

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.

2. From the first menu of the SMIT Interface, select `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Security.`
6. From the next SMIT menu, select `Add SNA Password.`
7. This displays a dialog that requests the new password. Enter the new password in accordance with the instructions, and press Enter (`Enter=Do`).
8. Again a dialog requests the new password. Reenter the new password, and press Enter to add the password.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
mksnapw
```

Deleting a Communication Authority Password

The following procedure explains how to delete a communication authority password, using the System Management Interface Tool (SMIT).

When the node security type is changed from currently secured to currently unsecured, the password is deleted.

The following procedure describes how to delete a communication authority password, using the SMIT Interface.

Deleting a Communication Authority Password

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Configure SNA Profiles`.
5. From the next SMIT menu, select `Security`.
6. From the next SMIT menu, select `Remove SNA Password`.
7. This displays a panel that requests the current password. Enter the password in accordance with the instructions, and press Enter (`Enter=Do`) to delete the password.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
rmsnapw
```

Changing a BIND Password

The following procedure explains how to change a BIND password, using the System Management Interface Tool (SMIT).

The following procedure describes how to change a BIND password, using the SMIT Interface.

Changing a BIND Password

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select

   `Communications Applications and Services.`
3. From the next SMIT menu, select `SNA Services.`
4. From the next SMIT menu, select `Configure SNA Profiles.`
5. From the next SMIT menu, select `Security.`
6. From the next SMIT menu, select `Change a BIND Password.`
7. This displays a name select dialog requesting the `LU type 6.2 CONNECTION profile name`. If you do not know the connection profile name, press F4 (F4=List) to display a list of all known connections. Select the connection for which the changed BIND password is desired, and press Enter (Enter=Do).
8. This displays a pop-up panel. Enter the new password and then reenter the password to confirm that the BIND password has been entered correctly. Press Enter to add the changed password.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
chsnaobj –t connection –u lu6.2 –v yes ProfileName
```

Generating 16-Character Hexadecimal Security Keys

The following procedure explains how to generate a 16-character hexadecimal security key, using the System Management Interface Tool (SMIT).

The Generate Security Keys option can generate the 16-character hexadecimal values used for BIND passwords. Refer to Node Verification in Defining LU Type 6.2 Connection Characteristics on page 13–185 for more information on the BIND password. The Generate Security Keys option is a tool. You can use it whenever you need to generate a hexadecimal key. This option is not necessary to define BIND passwords.

The following procedure describes how to generate security keys, using the SMIT Interface.

Generating Security Keys

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.

2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Configure SNA Profiles`.
5. From the next SMIT menu, select `Security`.
6. From the next SMIT menu, select `Generate Security Keys`.
7. This displays the `Generate Security Keys Dialog`. Type in the 30 to 80 character phrase and the number of keys to be generated, and press Enter (`Enter=Do`).

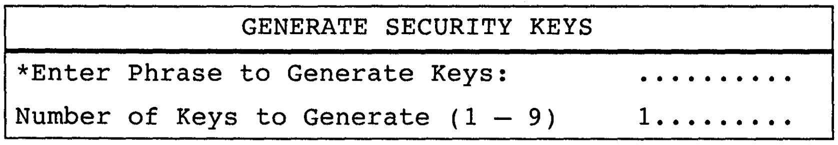
Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time the SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
genkeys –n Number Phrase
```

Generate Security Keys Dialog



```
                  GENERATE SECURITY KEYS
*Enter Phrase to Generate Keys:               ..........
Number of Keys to Generate (1 – 9)            1.........
```

Figure 1. Generate Security Keys Dialog

- Idle
- Test pending
- Testing.

You can get the status information for AIX SNA Services/6000 or a particular connection or attachment through the SMIT Interface or the SRC commands.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information about the SMIT Interface. Refer to the System Resource Controller Overview in *General Concepts and Procedures* for more information about the SRC.

Getting Connection Status

The AIX SNA Services/6000 subsystem returns data indicating the state of the connection profile.

The following procedure describes how to show the status of a connection, using the SMIT Interface.

Getting Status of the Connection

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.

2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Control SNA Services`.
5. From the next SMIT menu, select `Show the Status of SNA Services`.
6. From the next SMIT menu, select `Show the Status of an SNA Connection`.
7. This displays a name select dialog asking for the `Connection Profile Name`. If you do not know the connection profile name, press the F4 key (`F4=List`) to display a list of all known connections. Select the connection for which status is desired and press Enter (`Enter=Do`).

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
lssrc –l –t connection –o ProfName
```

Getting Attachment Status

The AIX SNA Services/6000 subsystem returns data indicating the state of the attachment profile.

The following procedure describes how to show the status of an attachment, using the SMIT Interface.

Getting Status of the Attachment

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Control SNA Services`.
5. From the next SMIT menu, select `Show the Status of SNA Services`.
6. From the next SMIT menu, select `Show the Status of an SNA Attachment`.
7. This displays a name select dialog asking for the `Attachment Profile Name`. If you do not know the attachment profile name, press the F4 key (`F4=List`) to display a list of all known attachments. Select the attachment for which status is desired and press Enter (`Enter=Do`).

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
lssrc –l –t attachment –o ProfName
```

A sample output appears on the screen. The status of an attachment contains information similar to the following panel. The information may exceed the size of the panel.

```
elist                Attachment      – active
cptest1              Control Point   – active
elist                Logical Link    – opened
elink                Physical Link   – opened
Ethernet             Logical Link Type
Ethernet             Physical Link Type
LEFT                 Local Network Name
RIGHT                Remote Network Name
————————————————— DATA LINK STATISTICS —————————————————
                   4 Remote SAP Address (hex)
                   0 Test Commands Sent
                   0 Test Command Failures
                   0 Test Commands Received
```

Tracing Network Activities

Attachment Test and Trace

An attachment test is a predefined message sent to a remote system, using a test protocol to determine if the attachment is working properly. The TRACE command activates or deactivates tracing of network activities for AIX SNA Services/6000 or a selected link attachment. When an attachment trace is activated for a specific attachment, the system logs link activities as they occur. Link activities that are logged include:

- Time outs
- Send data
- Receive data
- Link opens
- Link closes.

The system can log activities on as many as seven different data link attachments at the same time.

SNA API Trace

This trace logs the entry and exit points of the SNA API routines, detailing the API commands executed by the application.

SNA Internal Errors

This facility shows SNA internal error logs. It handles SNA activities such as attachment and connection failures, protocol violations, and configuration problems. When SNA internal error logging is activated for AIX SNA Services/6000, the system logs SNA activities as they occur. SNA activities that are logged are:

- Attachment failures
- Connection failures
- Protocol violations.

SNA System Errors

This facility shows SNA system error logs, detailing errors SNA has encountered with operating system functions.

Starting an Attachment Trace

The following procedure explains how to start an attachment trace, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `Attachment Trace`.
6. From the next SMIT menu, select `Start Trace`.
7. This displays a name select dialog asking for the `ATTACHMENT profile name`. If you do not know the attachment profile name, press the F4 key (`F4=List`) to display a list of all known attachments. Select the attachment you wish to start and press Enter (`Enter=Do`).

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this dialog is:

```
traceson –l –t attachment –o ProfName
```

Stopping an Attachment Trace

The following procedure explains how to stop an attachment trace, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `Attachment Trace`.
6. From the next SMIT menu, select `Stop Trace`.
7. This displays a name select dialog asking for the `ATTACHMENT profile name`. If you do not know the attachment profile name, press the F4 key (`F4=List`) to display a list of

Starting an API Trace

The following procedure explains how to start an API trace, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `SNA API Trace`.
6. From the next SMIT menu, select `Start Trace`. This starts an SNA API trace.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this procedure is:

```
trace –a –j 271
```

Stopping an API Trace

The following procedure explains how to stop an API trace, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `SNA API Trace`.
6. From the next SMIT menu, select `Stop Trace`. This stops an SNA API trace.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this procedure is:

```
trcstop
```

Starting an SNA Internal Error Trace

The following procedure explains how to start an SNA internal error trace, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `SNA Internal Errors`.
6. From the next SMIT menu, select `Start Trace`. This starts an SNA internal error trace.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this procedure is:

```
traceson -s sna
```

Stopping an SNA Internal Error Trace

The following procedure explains how to stop an SNA internal error trace, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `SNA Internal Errors`.
6. From the next SMIT menu, select `Stop Trace`. This stops an SNA internal error trace.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this procedure is:

```
tracesoff -s sna
```

Clearing an SNA System Error Log

The following procedure explains how to clear an SNA system error log, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `SNA System Errors`.
6. From the next SMIT menu, select `Clear Error Log`. This clears the SNA system error log.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

Generated Command

Any time a SMIT dialog is displayed, you may select the F6 key (`F6=Command`) to show the generated command. Press Enter (`Enter=Do`) to issue the command. The generated command for this procedure is:

```
errclear 0
```

Showing an SNA System Error Log

The following procedure explains how to show an SNA system error log, using the System Management Interface Tool (SMIT).

1. Start the System Management Interface Tool (SMIT) by entering the following command on the AIX command line:

   ```
   smit
   ```

   Entering `smit sna` takes you directly to step 4.
2. From the first menu of the SMIT Interface, select `Communications Applications and Services`.
3. From the next SMIT menu, select `SNA Services`.
4. From the next SMIT menu, select `Diagnose SNA Services`.
5. From the next SMIT menu, select `SNA System Errors`.
6. From the next SMIT menu, select `Show Error Log`. This shows the SNA system error log.

Refer to the System Management Interface Tool (SMIT) Overview in *General Concepts and Procedures* for more information on the SMIT Interface.

AIX SNA Services/6000 Commands

AIX SNA Services/6000 Commands describes the SNA Services/6000 commands and the Object Classes associated with the **mksnaobj** and **chsnaobj** commands. The following commands are available when AIX SNA Services/6000 is installed on your system:

The **chsnalias** Command

The **chsnaobj** Command

The **chsnapw** Command

The **exportsna** Command

The **gensnakey** Command

The **importsna** Command

The **linktest** Command

The **lssnaobj** Command

The **mksnalias** Command

The **mksnaobj** Command

The **mksnapw** Command

The **peu** Command

The **qrysnaobj** Command

The **rmsnalias** Command

The **rmsnaobj** Command

The **rmsnapw** Command

The **sna_update.awk** Command

The **verifysna** Command

The Object Classes section describes the flags that can be used with the **mksnaobj** and **chsnaobj** commands to create and change SNA Services/6000 profiles.

Add SNA Profile

Note: See Defining SNA Services Characteristics on page 13–211 for descriptions of the required information.

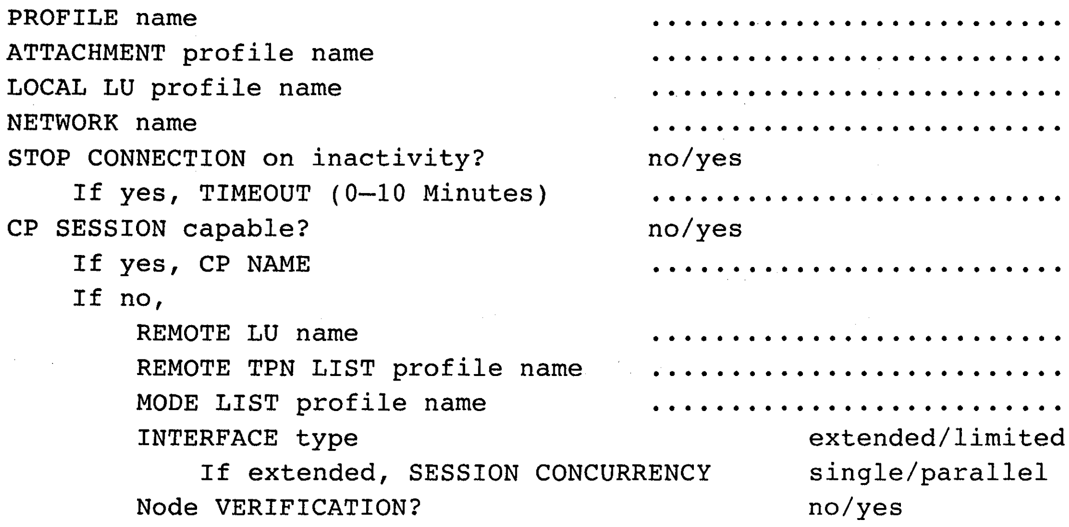
```
PROFILE name                                  .........................
Total active open CONNECTIONS (1–999)         .........................
Total SESSIONS (1–999)                        .........................
Total CONVERSATIONS (1–999)                   .........................
SERVER synonym name                           .........................
RESTART action                                once/respawn
Perform ERROR LOGGING?                        no/yes
Standard INPUT file/device                    .........................
Standard OUTPUT file/device                   .........................
Standard ERROR file/device                    .........................
```

Figure 1. Add SNA Profile Dialog

The base command for this task is: **mksnaobj –t sna** *ProfileName.*

Add Logical Unit Type 6.2 Connection Profile

Note: See Defining LU Type 6.2 Connection Characteristics on page 13–181 for descriptions of the required information.



```
PROFILE name                                    ..........................
ATTACHMENT profile name                         ..........................
LOCAL LU profile name                           ..........................
NETWORK name                                    ..........................
STOP CONNECTION on inactivity?                  no/yes
    If yes, TIMEOUT (0–10 Minutes)              ..........................
CP SESSION capable?                             no/yes
    If yes, CP NAME                             ..........................
    If no,
        REMOTE LU name                          ..........................
        REMOTE TPN LIST profile name            ..........................
        MODE LIST profile name                  ..........................
        INTERFACE type                                    extended/limited
            If extended, SESSION CONCURRENCY              single/parallel
        Node VERIFICATION?                                no/yes
```

Figure 3. Add LU Type 6.2 Connection Profile Dialog

The base command for this task is: **mksnaobj –t connection –u lu6.2** *ProfileName.*

Add Logical Unit Type 6.2 Local LU Profile

Note: See Defining LU Type 6.2 Local Logical Unit Characteristics on page 13–177 for descriptions of the required information.

```
PROFILE name                                      ..........................
TPN LIST profile name                             ..........................
CP SESSION capable?                               no/yes
    If no,
        NETWORK name                              ..........................
        Local LU NAME                             ..........................
INDEPENDENT LU?                                   no/yes
    If no,
        Local LU ADDRESS                          ..........................
        SSCP ID                                   ..........................
```

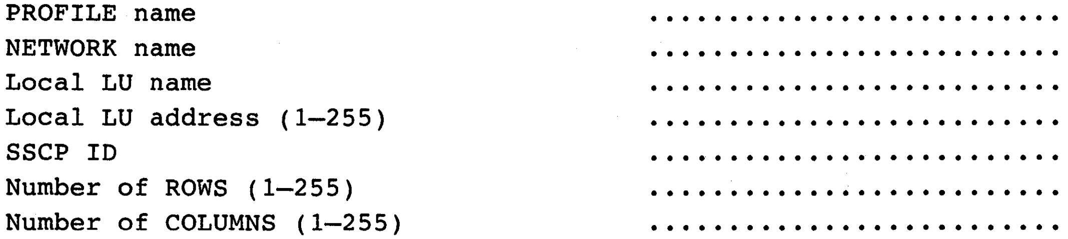
Figure 5. Add LU Type 6.2 Local Logical Unit Profile Dialog

The base command for this task is: **mksnaobj –t local_lu –u lu6.2** *ProfileName*.

Add Logical Unit Type x (2,3) Local LU Profile

Note: See the following for descriptions of the required information:

- Defining LU Type 2 Local Logical Unit Characteristics on page 13–174
- Defining LU Type 3 Local Logical Unit Characteristics on page 13–174.



```
PROFILE name                                  .........................
NETWORK name                                  .........................
Local LU name                                 .........................
Local LU address (1–255)                      .........................
SSCP ID                                       .........................
Number of ROWS (1–255)                        .........................
Number of COLUMNS (1–255)                     .........................
```

Figure 7. Add LU Type x (2,3) Local Logical Unit Profile Dialog

The base command for this task is: **mksnaobj –t local_lu –u lu***x ProfileName.*

Add Logical Unit Type 6.2 Mode List Profile

Note: See Defining LU Type 6.2 Modes for a Session on page13–191 for descriptions of the required information.

```
PROFILE name                                  ..........................
Add profile names to list:
    Name 1                                    ..........................
    Name 2                                    ..........................
    Name 3                                    ..........................
    Name 64                                   ..........................
```

Figure 9. Add LU Type 6.2 Mode List Profile Dialog

The base command for this task is: **mksnaobj –t mode_list** *ProfileName.*

Add Logical Unit Type 6.2 TPN List Profile

Note: See Defining LU Type 6.2 Transaction Programs for a Session on page 13–201 for descriptions of the required information.

```
PROFILE name                                    ..........................
Add profile names to list:
    Name 1                                      ..........................
    Name 2                                      ..........................
    Name 3                                      ..........................
    Name 64                                     ..........................
```

Figure 11. Add LU Type 6.2 TPN List Profile Dialog

The base command for this task is: **mksnaobj –t tpn_list** *ProfileName*.

Add Logical Unit Type 6.2 RTPN List Profile

Note: See Defining LU Type 6.2 Remote Transaction Programs for a Session on page 13–208 for descriptions of the required information.

```
PROFILE name                                    .........................
Add profile names to list:
    Name 1                                      .........................
    Name 2                                      .........................
    Name 3                                      .........................
    Name 64                                     .........................
```

Figure 13. Add LU Type 6.2 RTPN List Profile Dialog

The base command for this task is: **mksnaobj –t rtpn_list** *ProfileName*.

Add Token-Ring Attachment Profile

Note: See Defining Token-Ring Attachment Characteristics on page 13–135 for descriptions of the required information.

```
PROFILE name                                ..........................
CONTROL POINT profile name                  ..........................
LOGICAL LINK profile name                   ..........................
PHYSICAL LINK profile name                  ..........................
STOP ATTACHMENT on inactivity?              no/yes
    If yes, inactivity TIMEOUT (0–10 minutes) ....................
LU address REGISTRATION?                    no/yes
    If yes, LU address REGISTRATION PROFILE name .................
CALL Type                                   listen/call
    If listen,
        AUTO-LISTEN?                        no/yes
        MINIMUM SAP Address (0x04 – 0xEC) ........................
        MAXIMUM SAP Address (0x04 – 0xEC) ........................
    If call, ACCESS ROUTING                 link_name/link_address
        If link-name, REMOTE LINK name      ..........................
        If link-address,
            Remote LINK address             ..........................
            Remote SAP address (0x04 – 0xEC) .....................
```

Figure 15. Add Token-Ring Attachment Profile Dialog

The base command for this task is: **mksnaobj –t attachment –w token_ring** *ProfileName.*

Add X.25 Attachment Profile

Note: See Defining X.25 Attachment Characteristics on page 13–149 for descriptions of the required information.

```
PROFILE name                            ..........................
CONTROL POINT profile name              ..........................
LOGICAL LINK profile name               ..........................
PHYSICAL LINK profile name              ..........................
STOP ATTACHMENT on inactivity?          no/yes
    If yes, inactivity TIMEOUT (0–10 Minutes) .................
LU address REGISTRATION?                no/yes
    If yes, LU address REGISTRATION PROFILE name .................
X.25 LEVEL                              1984/1980
CALL Type                               listen/call
    If listen,
        AUTO-LISTEN?                    no/yes
        LISTEN NAME                     ..........................
    If call, VIRTUAL CIRCUIT type       switched/permanent
        If permanent,
            Logical CHANNEL number of PVC (1–4095) ..............
        If switched,
            Remote station X.25 address ..........................
            Optional X.25 facilities?   no/yes
                If yes,
                    REVERSE CHARGING?   no/yes
                    RPOA?               no/yes
                        If yes, DATA NETWORK ID codes ...........
                    PACKET size for RECEIVED data ................
                    PACKET size for TRANSMIT data ................
                    WINDOW size for RECEIVED data ................
                    WINDOW size for TRANSMIT data ................
                    THROUGHPUT Class for RECEIVED data ...........
                    THROUGHPUT Class for TRANSMIT data ...........
                    CLOSED USER group?                      no/yes
                        If yes, INDEX to closed group ............
                    Closed user group with OUTGOING ACCESS? no/yes
                        If yes, INDEX to closed group ............
                    Network USER IDentification?            no/yes
                        If yes, network USER ID name ............
```

Figure 17. Add X.25 Attachment Profile Dialog

The base command for this task is: **mksnaobj –t attachment –w x.25** *ProfileName*.

Add SDLC Secondary Logical Link Profile

Note: See Defining SDLC Secondary Logical Link Characteristics on page 13–63 for descriptions of the required information.

```
PROFILE name                                     ..........................
PHYSICAL LINK type                               ..........................
TRANSMIT window count                            ..........................
RETRANSMIT count (1–50)                          ..........................
Retransmit THRESHOLD (0–100)                     ..........................
DROP LINK on inactivity?                         no/yes
FORCE DISCONNECT timeout (1–600 seconds) ........................
DEFINITION of maximum I-FIELD size      system_defined/user_defined
    If user-defined, max. I-FIELD SIZE (265–30729) ...............
TRACE Link?                                      no/yes
    If yes, TRACE SIZE                           short/long
Secondary INACTIVITY timeout (1–120 seconds) .....................
Local SECONDARY STATION address                  ..........................
```

Figure 19. Add SDLC Secondary Logical Link Profile Dialog

The base command for this task is: **mksnaobj –t log_sdlc –b secondary** *ProfileName*.

Add Standard Ethernet Logical Link Profile

Note: See Defining Standard Ethernet Logical Link Characteristics on page 13–113 for descriptions of the required information.

```
PROFILE name                                      ..........................
TRANSMIT window count (1–127)                     ..........................
RETRANSMIT count (1–30)                           ..........................
RECEIVE window count (1–127)                      ..........................
DROP LINK on inactivity?                          yes/no
INACTIVITY timeout (1–120 seconds)                ..........................
RESPONSE timeout (1–40, 500 msec intervals) ......................
ACKNOWLEDGE timeout (1–40, 500 msec intervals) ..................
FORCE DISCONNECT timeout (1–600 seconds) ........................
DEFINITION of maximum I-FIELD size               system_defined/user_defined
    If user-defined, max. I-FIELD SIZE (265–30729) ...............
TRACE Link?                                       no/yes
    If yes, TRACE SIZE                            short/long
```

Figure 21. Add Standard Ethernet Logical Link Profile Dialog

The base command for this task is: **mksnaobj –t log_ethnet** *ProfileName*.

Add Token-Ring Logical Link Profile

Note: See Defining Token-Ring Logical Link Characteristics on page 13–140 for descriptions of the required information.

```
PROFILE name                                ..........................
TRANSMIT window count (1–127)               ..........................
DYNAMIC window increment (1–127)            ..........................
RETRANSMIT count (1–30)                     ..........................
RECEIVE window count (1–127)                ..........................
RING ACCESS priority                        ..........................
DROP LINK on inactivity?                   yes/no
INACTIVITY timeout (1–120 seconds)          ..........................
RESPONSE timeout (1–40, 500 msec intervals) .......................
ACKNOWLEDGE timeout (1–40, 500 msec intervals) ...................
FORCE DISCONNECT timeout (1–600 seconds) ........................
DEFINITION of maximum I-FIELD size     system_defined/user_defined
    If user-defined, max. I-FIELD SIZE (265–30729) ...............
TRACE Link?                                 no/yes
    If yes, TRACE SIZE                      short/long
```

Figure 23. Add Token-Ring Logical Link Profile Dialog

The base command for this task is: **mksnaobj –t log_toknrng** *ProfileName*.

Add IEEE 802.3 Ethernet Logical Link Profile

Note: See Defining IEEE 802.3 Ethernet Logical Link Characteristics on page 13–126 for descriptions of the required information.

```
PROFILE name                                .........................
TRANSMIT window count (1–127)               .........................
RETRANSMIT count (1–30)                     .........................
RECEIVE window count (1–127)                .........................
DROP LINK on inactivity?                    yes/no
INACTIVITY timeout (1–120 seconds)          .........................
RESPONSE timeout (1–40, 500 msec intervals) .....................
ACKNOWLEDGE timeout (1–40, 500 msec intervals) ..................
FORCE DISCONNECT timeout (1–600 seconds) ........................
DEFINITION of maximum I-FIELD size         system_defined/user_defined
    If user-defined, max. I-FIELD SIZE (265–30729) ..............
TRACE Link?                                 no/yes
    If yes, TRACE SIZE                      short/long
```

Figure 25. Add IEEE 802.3 Ethernet Logical Link Profile Dialog

The base command for this task is: **mksnaobj –t log_802.3** *ProfileName*.

Add QLLC Logical Link Profile

Note: See Defining QLLC Logical Link Characteristics on page 13–157 for descriptions of the required information.

```
PROFILE name                                    .........................
DROP LINK on inactivity?                        no/yes
FORCE DISCONNECT timeout (1–600 seconds) .........................
DEFINITION of maximum I-FIELD size       system_defined/user_defined
    If user-defined, max. I-FIELD SIZE (265–30729) ...............
TRACE Link?                                     no/yes
    If yes, TRACE SIZE                          short/long
STATION type                                    .........................
    If secondary or negotiable,
        Secondary INACTIVITY timeout            .........................
    If primary or negotiable,
        Primary repoll TIMEOUT (1–255 seconds) ..................
        Primary repoll COUNT (1–255)            .........................
```

Figure 27. Add QLLC Logical Link Profile Dialog

The base command for this task is: **mksnaobj –t log_x.25** *ProfileName*.

Add EIA232D Physical Link Profile

Note: See Defining EIA232D Physical Link Characteristics on page 13–81 for descriptions of the required information.

```
PROFILE name                                ..........................
DATALINK device name                        ..........................
Serial ENCODING                             NRZI/NRZ
Request to send (RTS)                       controlled/continuous
DTR control                                 DTR/CDSTL
Bit CLOCKING                                external/internal
    If external, DATA rate select           full/alternate
    If internal, TRANSMIT rate (600–38400) .......................
NETWORK type                                switched/nonswitched
    If switched, CALL type                  listen/call
        If listen,
            AUTO-LISTEN?                    no/yes
            CALL-OVERRIDE?                  no/yes
            ANSWER MODE                     automatic/manual
```

Figure 29. Add EIA232D Physical Link Profile Dialog

The base command for this task is: **mksnaobj –t phy_eia232d** *ProfileName.*

Add Smart Modem Physical Link Profile

Note: See Defining Smart Modem Physical Link Characteristics on page 13–86 for descriptions of the required information.

```
PROFILE name                                 ..........................
DATALINK device name                         ..........................
Serial ENCODING                              NRZI/NRZ
Request to send (RTS)                        controlled/continuous
DTR control                                  DTR/CDSTL
Dial DATA rate select                        full/alternate
Dial TRANSMIT rate (600—38400)               ..........................
NETWORK type                                 switched/nonswitched
    If switched, CALL type                   listen/call
        If listen,
            AUTO-LISTEN?                     no/yes
            CALL-OVERRIDE?                   no/yes
            ANSWER MODE                      automatic/manual
        If call,
            AUTO-CALL?                       no/yes
            Connect TIMER (50—600, .1 sec) ......................
```

Figure 31. Add Smart Modem Physical Link Profile Dialog

The base command for this task is: **mksnaobj –t phy_smtmdm** *ProfileName*.

Add EIA422A and V.35 Physical Link Profile

Note: See one of the following for descriptions of the required information:

- Defining EIA422A Physical Link Characteristics on page 13–96
- Defining V.35 Physical Link Characteristics on page 13–105.

```
PROFILE name                               .........................
DATALINK device name                       .........................
Serial ENCODING                            NRZI/NRZ
Request to send (RTS)                      controlled/continuous
DTR control                                DTR/CDSTL
Bit CLOCKING                               external/internal
    If external, DATA rate select          full/alternate
    If internal, TRANSMIT rate (600–38400) ......................
```

Figure 33. Add EIA422A and V.35 Physical Link Profile Dialog

The base command for this task is:

- **mksnaobj –t phy_eia422a** *ProfileName* or
- **mksnaobj –t phy_v.35** *ProfileName*

LU 6.2 Configurations

Each of the five LU 6.2 configurations discussed here requires 12 separate profiles:

- Connection
- Local LU
- Mode
- Mode List
- Transaction Program Name (TPN)
- TPN List
- Remote Transaction Program Name (RTPN)
- RTPN List
- Control Point
- Attachment
- Logical Link
- Physical Link.

AIX Node to AIX Node File Transfers

The profiles described in the **rcvtrn.prof** file are provided as an example of profiles that allows you to transfer files from one AIX node to another AIX node. They can be used in conjunction with transactions much like the **sendto.c** and **rcvfrom.c** sample programs described in Transferring Files in *Communications Programming Concepts*.

AIX SNA Services/6000 LU 6.2 File Transfers with Other LU 6.2 Systems

The following information relates to file transfers between an AIX Node and CICS or VTAM on a host, a System/36, or a System/38. Only the fields where a change is needed are discussed.

AIX SNA Services/6000 LU 6.2 to 6.2 Systems Running CICS

The profiles described in the **cics.prof** file are provided as an example of profiles that allow you to run cooperating applications between an AIX node and a host running CICS.

Note: You must modify some of the profile fields to meet your specific needs. Make sure that the necessary configuration fields match the values expected by your CICS host.

AIX SNA Services/6000 LU 6.2 to 6.2 Systems/36 or System/38

The profiles described in the **s36.prof** and **s38.prof** files are provided as examples of profiles that allow you to run cooperating applications between an AIX node and a System/36 or System/38 node, respectively.

Note: You must modify some of the profile fields to meet your specific needs. Make sure that the necessary configuration fields match the values expected by your hSystem/36 or System/38.

Matching Host and AIX SNA Services/6000 Configuration Parameters

As you assign values to AIX SNA Services/6000 configuration parameters for LU 1, LU 2, LU 3, and LU 6.2 dependent-type variables, be sure you assign the same values defined by the host system. The table that follows contains some of these variables, with examples for SDLC/EIA232D connections included in parentheses.

Host Keyword	SNA Configuration Profile	SNA Configuration Value
Remote LU Name (THX99602)	Connection	Remote LU Name/CP Name (THX99602)
DLOGMOD	Connection	LU/Session Type
DLOGMOD (D3270M2)	Local LU	LU/Session Type (LU2)
DLOGMOD (P3270L1)	Local LU	LU/Session Type (LU1)
DLOGMOD (P3270L3)	Local LU	LU/Session Type (LU3)
LOCADDR (7)	Local LU	Local LU Address (7)
SSCPID, in decimal (4672)	Local LU	SSCP ID, in hex (050000001240)
PUTYPE (2)	Attachment	Station Type (secondary)
IDBLK + ID NUM (071 + 00996)	Control Point PU	XID Node ID (07100996)
PUTYPE	SDLC Logical Link	Station Type (stationary)
NRZI (yes)	SDLC Logical Link	Serial Encoding (NRZI)
PU ADDRESS, in hex (C1)	SDLC Logical Link	Local Secondary Station Address (193, decimal)
CLOCKING (ext)	EIA232D Physical Link	Bit Clocking (external)
SPEED (2400)	EIA232D Physical Link	Transmission Rate (2400)
DUPLEX (Full)	EIA232D Physical Link	Data Rate Select (Full Duplex)
CALL (In)	EIA232D Physical Link	Call Type (Call)

TCP/IP Internet Terms

Protocols

In any communication environment, each host must follow certain rules (called *protocols*) that allow other hosts to receive and interpret messages sent to them. TCP/IP supports a *suite* of protocols, each of which provides a different service. These protocols are the mechanism that allows networking communication to be independent of the network hardware.

The TCP/IP protocol suite is organized into the following groups:

- Internet Application-Level Protocols
- Internet Transport-Level Protocols
- Internet Network-Level Protocols.

Packets and Datagrams

Information is sent over the Internet in small blocks called *packets*. Protocols break information into smaller chunks called *datagrams*, add headers, and then transmit the datagrams over the network to a destination. In this documentation, the term datagram is associated with the Internet layer protocols; the term packet is associated with the physical network layer. However, in the industry, these terms are sometimes used interchangeably.

Addresses

For reliable communication, each Internet host is assigned at least one unique Internet address. This 32-bit address is used by protocols for routing packets properly across a network that uses TCP/IP. Each network interface to the Internet network is assigned its own unique address. The machine specified as the gateway host between two or more networks may have more than one interface address. When a packet is transported over the network, the packet includes the Internet address of the source host as well as the destination host. Internet addresses identify both the network address and the local host address.

Names

Each computer, or host, on an Internet network is assigned at least one Internet host name and may have several aliases. In TCP/IP, names are translated, or *resolved,* into Internet addresses using the **/etc/hosts** file for a flat network or a name server in a *domain* network.

Routing

Routing allows information to be directed from a source host to a destination host. There are two types of routing in TCP/IP: *static routing* and *dynamic routing*. Static routes can be defined on each Internet host for common destinations. Routes can be defined dynamically by *routing daemons*, which find routes to destinations that have not been defined in the routing tables.

If you want two networks to communicate with each other, you can connect them through one machine, called a *gateway* machine. This machine must physically be on both networks. A gateway contains the addressing and routing information for each host on its network, and may use *routing daemons* to broadcast routing information to, and receive routing information from, other gateways. TCP/IP routes information to the appropriate computer on the network using address information carried in a packet or stream of information.

Message Delivery for TCP/IP

TCP/IP provides two types of message delivery for commands, services, and application programs: connectionless packet delivery and reliable stream delivery. These two data transport services provide the backbone of the Internet and TCP/IP.

For further information about TCP/IP, read Network Management Commands on page 14–7, Understanding Interfaces for TCP/IP on page 14–45, Understanding Addresses for TCP/IP on page 14–30, Understanding Naming for TCP/IP on page 14–24, Understanding Packets for TCP/IP on page 14–43, Understanding Protocols for TCP/IP on page 14–54, Understanding Routing for TCP/IP on page 14–37, and Understanding Gateways for TCP/IP on page 14–39.

Related Information

Networks for System Management Overview on page 5–1.

Sockets Overview in *General Programming Concepts*.

Understanding Basic Functions of TCP/IP

TCP/IP includes commands and facilities that provide the basic functions available to end users and application programs. These functions include:

- File transfer
- Remote mail and interactive conversation
- Remote login, command execution, and printing
- Network management.

File Transfer

TCP/IP contains three file transfer commands:

ftp Uses the FTP protocol to transfer files between hosts that use dissimilar file systems. The **ftp** command provides subcommands that allow changing the current local and remote directory, transferring multiple files in a single request, creating and removing directories, and escaping to the local shell to perform shell commands. It provides for security by sending passwords to the remote host and also permits automatic login, file transfers, and logoff.

rcp Copies one or more files between the local host and a remote host, between two remote hosts, or between files at the same remote host.

tftp (utftp) Uses the Trivial File Transfer Protocol (TFTP) to transfer files to and from hosts. Since TFTP is a minimal file transfer protocol, the **tftp** and **utftp** commands do not provide all the features of the **ftp** command.

Remote Mail and Conversations

TCP/IP provides the **talk** command, which allows two users on the same host or different hosts to have an interactive conversation.

TCP/IP also provides the **mail** and **sendmail** commands, as well as the commands included with the MH package.

send and receive data through Internet networks. A packet is sent from a *source* to a *destination.*

port A logical connecting point for a process. Data is transmitted between processes through ports (or *sockets*). Each port provides queues for sending and receiving data. In an Interface Program network, each port has an Internet *port number* based on how it is being used. A particular port is identified with an Internet *socket address,* which is the combination of an Internet host address and a port number.

process A program that is running. A process is the active element in a computer. Terminals, files, and other I/O devices communicate with each other through processes. Thus, network communication is *interprocess communication* (that is, communication between processes).

protocol A set of rules for handling communications at the physical or logical level. Protocols often use other protocols to provide services. For example, a *connection-level protocol* uses a *transport-level protocol* to transport packets that maintain a connection between two hosts.

server A computer or process that provides data, services, or resources that can be accessed by other computers or processes on the network.

For further information on TCP/IP system management, read Understanding Basic Functions of TCP/IP on page 14–4.

For further information on Internet topics, read Understanding Protocols for TCP/IP on page 14–54, Understanding Routing for TCP/IP on page 14–37, and Understanding the TCP/IP Daemons on page 14–19.

Related Information

Planning Your TCP/IP Network on page 14–94.

TCP/IP Daemons

fingerd The **fingerd** daemon provides remote user information.

ftpd The **ftpd** daemon provides the server function for the DARPA Internet File Transfer Protocol (FTP).

gated The **gated** daemon provides gateway routine functions and supports the Routing Information Protocol (RIP), Exterior Gateway Protocol (EGP), DCN Local-Network Protocol (HELLO), and the Simple Network Management Protocol (SNMP).

inetd The **inetd** command provides intermediate service management.

named The **named** daemon provides the server function for the Domain Name Protocol (DOMAIN).

rexecd The **rexecd** daemon provides the server function for the **rexec** command.

rlogind The **rlogind** daemon provides the server function for the **rlogin** command.

routed The **routed** daemon manages network routing tables and supports the RIP protocol.

rshd The **rshd** daemon provides the server function for remote command execution.

rwhod The **rwhod** daemon provides the server function for the **rwho** and **ruptime** commands.

syslogd The **syslogd** daemon reads and logs system messages.

talkd The **talkd** daemon provides the server function for the **talk** command.

telnetd The **telnetd** daemon provides the server function for the TELNET protocol.

tftpd The **tftpd** daemon provides the server function for the Trivial File Transfer Protocol.

timed The **timed** daemon accesses the time service daemon.

For further information on daemons, read Understanding the TCP/IP Daemons on page 14–19.

Related Information

System Resource Controller Overview in *General Programming Concepts.*

Auditing

Network auditing is provided by TCP/IP, using the AIX **audit** subsystem to audit both kernel network routines and application programs. The purpose of auditing is to record those actions that affect the security of the system and the user responsible for those actions.

The following types of events are audited:

Kernel Events

- Change configuration
- Change host ID
- Change route
- Connection
- Create socket
- Export object
- Import object.

Application Events

- Access the network
- Change configuration
- Change host ID
- Change static route
- Configure mail
- Connection
- Export data
- Import data
- Write mail to a file.

Creation and deletion of objects are audited by AIX. Application audit records suspend and resume auditing to avoid redundant auditing by the kernel.

Network Interfaces

You can establish security not just for a host, but for each network interface attached to that host. This adds flexibility to the security features over securing a system by operation or by host. Each network interface can operate at a different level of security, and the users for each network interface on a host can operate at different authority levels for each interface. For instance, user Joe can log in to host Funky and use three different network interfaces. Joe can have `genser` authority on the `tr0` interface, `siop` authority on the `en0` interface, and `ndsccs-criticom` authority on the `en1` interface.

Two network interfaces must operate at the same security level to communicate. If two users are set up with authority levels, their authority levels must be the same, as well. Security levels are set by default during automatic configuration of network interfaces. To customize the settings, use the System Management Interface Tool (SMIT) or the **ifconfig** command.

For further information on security, read Understanding Trusted Processes on page 14–15, Understanding TCP/IP Command Security on page 14–13, Understanding the Network Trusted Computing Base on page 14–16, and Understanding Data Security and Information Protection on page 14–18.

Related Information

The **ftpd** daemon, **rexecd** daemon, **telnetd** daemon.

The **securetcpip** command, **ifconfig** command.

The **audit** command, **ftp** command, **tftp** command, **rexec** command, **tsh** command, **telnet** command.

Planning Your TCP/IP Network on page 14–94.

RFC 1038, *IP Revised Security Option.*

Note: Since use of the **.netrc** file requires storage of passwords in a non-encrypted file, the automatic login feature of the **ftp** command is not available when your system has been configured with the **securetcpip** command. This feature can be re-enabled by removing the **ftp** command from the tcpip: stanza in the **/etc/security/config** file.

In order to use the file transfer function, the **ftp** command requires two TCP/IP connections, one for the protocol and one for data transfer. The protocol connection is primary and is secure because it is established on reliable communicating ports. The secondary connection is needed for the actual transfer of data, and both the local and remote host verify that the other end of this connection is established with the same host as the primary connection. If the primary and secondary connections are not established with the same host, the **ftp** command first displays an error message stating that the data connection was not authenticated and then exits. This verification of the secondary connection prevents a third host from intercepting data intended for another host.

rexec The **rexec** command provides a secure environment for executing commands on a foreign host. The user is prompted for both a login ID and a password.

An automatic login feature causes the **rexec** command to search the **$HOME/.netrc** file for the user's ID and password on a foreign host. For security, the permissions on the **$HOME/.netrc** file must be set to 600 (read and write by owner only). Otherwise, automatic login fails.

Note: Since use of the **.netrc** file requires storage of passwords in a non-encrypted file, the automatic login feature of **rexec** is not available when your system is operating in secure. This feature can be re-enabled by removing the **rexec** entry from the `tcpip:` stanza in the **/etc/security/config** file.

telnet or **tn** The **telnet** (TELNET) command provides a secure environment for login to a foreign host. The user is prompted for both a login ID and a password. The user's terminal is treated just like a terminal connected directly to the host. That is, access to the terminal is controlled by permission bits. Other users (group and other) do not have read access to the terminal, but they can write messages to it if the owner gives them write permission. The **telnet** command also provides access to a trusted shell on the remote system through the Secure Attention Key (SAK). This key sequence differs from the sequence that invokes the local trusted path and can be defined with the **telnet** command.

For further information on security, read Understanding Trusted Processes on page 14–15, Understanding Network Trusted Computing Base on page 14–16, and Understanding Data Security and Information Protection on page 14–18.

Related Information

The **ftpd** daemon, **rexec** daemon, **telnetd** daemon.

The **securetcpip** command, **ifconfig** command.

Understanding Interfaces for TCP/IP on page 14–45.

RFC 1038, *Revised IP Security Option.*

Understanding the Network Trusted Computing Base (NTCB)

The network contains both hardware and software mechanisms to implement the networking security features. This section defines the components of the Network Trusted Computing Base (NTCB) as they relate to TCP/IP.

The hardware security features for the network are provided by the network adapters used with TCP/IP. These adapters are programmed to control incoming data by receiving only data destined for the local system and to broadcast data receivable by all systems.

The software component of the NTCB consists of only those programs that are considered trusted. The programs and associated files that are part of a secure system are listed in the following sections on a directory-by-directory basis.

/etc				
Name	**Owner**	**Group**	**Mode**	**Permissions**
arp	root	system	4555	r–sr–xr–x
gated.conf	root	system	0664	rw–rw–r—
gateways	root	system	0664	rw–rw–r—
hosts	root	system	0664	rw–rw–r—
hosts.equiv	root	system	0664	rw–rw–r—
ifconfig	bin	bin	0555	r–xr–xr–x
inetd.conf	root	system	0664	rw–rw–r—
named.boot	root	system	0664	rw–rw–r—
named.data	root	system	0664	rw–rw–r—
networks	root	system	0664	rw–rw–r—
ping	root	system	4555	r–sr–xr–x
protocols	root	system	0664	rw–rw–r—
rc.tcpip	root	system	0774	rwxrwxr—
resolv.conf	root	system	0664	rw–rw–r—
route	root	system	4554	r–sr–xr—
securetcpip	root	system	4554	r–sr–xr—
services	root	system	0664	rw–rw–r—
syslogd	root	system	4554	r–sr–xr—
3270.keys	root	system	0664	rw–rw–r—
3270keys.rt	root	system	0664	rw–rw–r—
fingerd	root	system	4554	r–sr–xr—
ftpd	root	system	4554	r–sr–xr—
gated	root	system	4554	r–sr–xr—
inetd	root	system	4554	r–sr–xr—
named	root	system	4554	r–sr–xr—
rexecd	root	system	4554	r–sr–xr—

Understanding Data Security and Information Protection

The security feature for TCP/IP does not encrypt user data transmitted through the network. Therefore, it is suggested that users identify any risk in communication that could result in the disclosure of passwords and other sensitive information, and based on that risk, apply appropriate countermeasures.

The use of this product in a Department of Defense (DOD) environment may require adherence to DOD 5200.5 and NCSD–11 for communications security.

The **ifconfig** command provides the IP security option to comply with the DOD Basic Security section of RFC 1038, *Revised IP Security Option.* Additional information, including addresses, for accrediting authorities whose protection rules apply at each level of security is available in RFC 1038.

For further information on security, read Understanding Trusted Processes on page 14–15, Understanding TCP/IP Command Security on page 14–13, and Understanding Network Trusted Computing Base on page 14–16.

Related Information

The **ifconfig** command.

RFC 1038, *Revised IP Security Option.*

routed Manages the network routing tables and supports the Routing Information Protocol (RIP). The **gated** daemon is preferred over the **routed** daemon, because the **gated** daemon supports more gateway protocols.

rshd Provides the remote command execution server function for the **rcp** and **rsh** commands.

rwhod Sends broadcasts to all other hosts every three minutes and stores information about logged-in users and network status. Use the **rwhod** daemon with extreme care, as it can steal significant amounts of a machine's resources.

syslogd Reads and logs system messages. This daemon is in the RAS group of subsystems.

talkd Provides the conversation function for the **talk** command.

telnetd Provides the server function for the DARPA standard TELNET protocol.

tftpd Provides the function for the Trivial File Transfer Protocol (TFTP).

timed Invokes the time-server daemon at system startup.

For further information on the TCP/IP daemons, read SRC Control of TCP/IP Daemons on page 14–21.

For further information on daemons and protocols, read Understanding the Domain Protocol (DOMAIN) on page 14–71, Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the File Transfer Protocol (FTP) on page 14–74, Understanding the Finger Protocol (FINGER) on page 14–76, Routing Information Protocol (RIP) on page 14–78, Understanding the TELNET Protocol on page 14–75, and Understanding the Trivial File Transfer Protocol (TFTP) on page 14–76.

Related Information

The **rcp** command, **rexec** command, **rlogin** command, **rsh** command, **talk** command.

The **syslogd** daemon.

The **rc.tcpip** file.

Subsystems and Subservers

A *subsystem* is a daemon, or server, that is controlled by SRC. A *subserver* is a daemon that is controlled by a subsystem. The only TCP/IP subsystem that controls other daemons is the **inetd** daemon. Thus, all TCP/IP subservers are also **inetd** subservers. See the above list for all TCP/IP subservers.

The categories of subsystem and subserver are mutually exclusive. That is, daemons are not listed as both a subsystem and as a subserver.

SRC Commands

SRC commands can affect one daemon, a group of daemons, or a daemon and those daemons it controls (subsystem with subservers). In addition, some TCP/IP daemons do not respond to all SRC commands. The following is a list of SRC commands that can be used to control TCP/IP daemons, and their exceptions.

startsrc Starts all TCP/IP subsystems and **inetd** subservers. The **startsrc** command works for all TCP/IP subsystems and **inetd** subservers.

stopsrc Stops all TCP/IP subsystems and **inetd** subservers. This command is also called the **stop normal**. The **stop normal** command allows subsystems to process all outstanding work and terminate gracefully. For **inetd** subservers, all pending connections are allowed to be started and all existing connections are allowed to be completed. The **stop normal** command works for all TCP/IP subsystems and **inetd** subservers.

stopsrc –f Stops all TCP/IP subsystems and **inetd** subservers. This command is also called the **stop force**. The **stop force** command immediately terminates all subsystems. For **inetd** subservers, all pending connections and existing connections are terminated immediately. The **stop force** command works for all TCP/IP subsystems and **inetd** subservers.

refresh Refreshes the following subsystems and subservers: the **inetd**, **syslogd**, **named**, and **gated** subsystems.

lssrc Provides short status for subsystems, which is the state of the specified subsystem (active or inoperative). Also provides short status for **inetd** subservers. The short status for **inetd** subservers includes: subserver name, state, subserver description, command name, and the arguments with which it was invoked.

lssrc –l Provides the short status plus additional information (long status) for the following subsystems:

gated State of debug or trace

Routing protocols activated

Routing tables

Signals accepted and their function.

inetd State of debug

InetServ object class

List of subservers and their short status

Signals accepted and their function.

Assigned Numbers Overview

For compatibility with the general network environment, well-known numbers are assigned for the Internet versions, networks, ports, protocols, and protocol options. Additionally, well-known names are also assigned to machines, networks, operating systems, protocols, services, and terminals. AIX TCP/IP complies with the assigned numbers and names defined in RFC 1010, *Assigned Numbers.*

The Internet Protocol defines a 4-bit field in the IP header that identifies the version of the general internetwork protocol in use. For IP, this version number in decimal is 4. For details on the assigned numbers and names used by TCP/IP, refer to the /**etc/protocols** and /**etc/services** files included with TCP/IP. For further details on the assigned numbers and names, refer to RFC 1010 and the /**etc/services** file.

For more information on protocols, see Understanding Protocols for TCP/IP on page 14–54.

Related Information

The /**etc/protocols** file, /**etc/services** file.

Understanding Naming for TCP/IP

Although 32-bit Internet addresses provide machines an efficient means of identifying the source and destination of datagrams sent across an internetwork, users prefer meaningful, easily remembered names. TCP/IP provides a naming system that supports both flat and hierarchical network organizations.

Naming in flat networks is very simple: host names consist of a single set of characters and generally are administered locally. In flat TCP/IP networks, each machine on the network has a file containing the name-to-Internet-address mapping information for every host on the network. As a TCP/IP network grows, the administrative burden of keeping each machine's naming file current grows. When TCP/IP networks become very large, as in the Internet, naming is divided hierarchically. Typically the divisions follow the network's organization. In TCP/IP, hierarchical naming is known as the *Domain* naming system and uses the DOMAIN protocol. The DOMAIN protocol is implemented by the **named** daemon in TCP/IP.

As in naming for flat networks, the domain name hierarchy provides for the assignment of symbolic names to networks and hosts that are meaningful and easy for users to remember. However, instead of each machine on the network keeping a file containing the name-to-address mapping for all other hosts on the network, one or more hosts are selected to function as *name servers.* Name servers translate (resolve) symbolic names assigned to networks and hosts into the efficient Internet addresses used by machines.

Note: Although AIX TCP/IP supports the flat naming convention, it only supports Domain name servers. If you want to use flat naming, each machine must have a current /**etc/hosts** file.

Naming Authority

In a flat network, all hosts in the network are administered by one central authority. This form of network requires that all hosts in the network have unique host names. In a large network, this requirement creates a large administrative burden on the central authority.

In a domain network, groups of hosts are administered separately within a tree-structured hierarchy of domains and subdomains. In this case, host names need to be unique only

Name Servers

In a flat name space, all names must be kept in the **/etc/hosts** file on each host on the network. If the network is very large, this can become a burden on the resources of each machine.

In a hierarchical network, certain hosts designated as *name servers* resolve names into Internet addresses for other hosts. This has two advantages over the flat name space: it keeps the resources of each host on the network from being tied up in resolving names, and it keeps the person who manages the system from having to maintain name resolution files on each machine on the network. The set of names managed by a single name server is known as its *zone of authority.*

Note: Although the host machine that performs the name resolution function for a zone of authority is commonly referred to as a *name server* host, the process controlling the function, the **named** daemon, is the actual name server process.

To further reduce unnecessary network activity, name servers *cache* (store for a period of time) name-to-address mappings. When a client asks a server to resolve a name, the server checks its cache first to see if the name has been resolved recently.

Within any autonomous system there can be multiple name servers. Typically, name servers are organized hierarchically and correspond to the network's organization. Referring to the previous diagram, each domain might have a name server responsible for all subdomains within the domain. Each subdomain name server communicates with the name server of the domain above it (called the *parent* name server), as well as with the name servers of other subdomains. For example, in the previous diagram, Austin, Palo Alto, and Raleigh are all subdomains of the domain IBM. If the tree hierarchy is followed in the network design, the Austin name server communicates with the name servers of Palo Alto and Raleigh as well as with the parent IBM name server. The Austin name server will also communicate with the name servers responsible for its subdomains.

There are several types of name servers.

Primary Name Server Loads its data from a file or disk and may delegate authority to other servers in its domain.

Secondary Name Server
Receives its information at system startup time for the given zone of authority from a primary name server, and then periodically asks the primary server to update its information.

Caching-Only Server All servers cache; that is, they cache the information that is received for use until the data expires (*time to live*, or TTL). A caching-only server responds to queries by askng other servers who have the authority to provide the information needed.

Forwarder or Client Server
Forwards queries it cannot satisfy locally to a fixed list of forwarding servers instead of interacting with the primary name servers for the root domain and other domains. The queries to the forwarding servers are recursive. There may be one or more forwarding servers, which are tried in turn until the list is exhausted. A client and forwarder configuration is typically used when you do not wish all the servers at a given site to interact with the rest of the Internet servers.

Note: If you are using network information service for name resolution, resolver routines will attempt to resolve names using the following sources in the order listed:

1. Network information service
2. The DOMAIN (**named**) name server
3. The local /**etc/hosts** file.

TCP/IP name servers use *caching* to reduce the cost of searching for names of hosts on remote networks. Instead of searching anew for a host name each time a request is made, a name server looks at its cache to see if the host name was resolved recently. Since domain and host names do change, each item remains in the cache for a limited length of time specified by the record's time to live (TTL). In this way, authorities can specify how long they expect the name resolution to be accurate.

In a DOMAIN name server environment, the host name set using the **hostname** command from the command line or in the **rc.net** file format must be the official name of the host as returned by the name server. Generally, this name is the full domain name of the host in the form:

```
host.subdomain.subdomain.rootdomain
```

If the host name is not set up as a fully qualified domain name, and if the system is set up to use a DOMAIN name server in conjunction with the **sendmail** program, the **sendmail** configuration file (/**usr/lib/sendmail.cf**) must be edited to reflect this official host name. In addition, the domain name macros in this configuration file must be set for the **sendmail** program to operate correctly.

Note: The domain specified in the /**usr/lib/sendmail.cf** file takes precedence over the domain set by the **hostname** command for all **sendmail** functions.

For a host that is in a domain network but is not a name server, the local domain name and the domain name server are specified in the /**etc/resolv.conf** file. In a DOMAIN name server host, the local domain and other name servers are defined in files read by the **named** daemon when it starts.

For further information on naming, read Planning for DOMAIN Name Resolution on page 14–108, Configuring Name Servers for TCP/IP on page 14–29, Understanding the Domain Name Protocol (DOMAIN) on page 14–71, and Understanding Addresses for TCP/IP on page 14–30.

Related Information

The **hostname** command, **sendmail** command.

The **named** daemon.

The **gethostbyname** subroutine, **gethostbyaddr** subroutine.

The **hosts** file, **rc.tcpip** file, **resolv.conf** file, **named.boot** file, **named Cache** file, **named Local** file, **named Data** file, **named Reverse Data** file.

The Standard Resource Record Format.

resolv.conf file   The presence of this file indicates to a host that it should go to a name server to resolve a name first. If the **resolv.conf** file does not exist, the host looks in the **/etc/hosts** file for name resolution. On a name server, the **resolv.conf** file should exist but be empty.

Time to live (TTL) is specified in resource records. If TTL is not specified in a record, the length of this time period defaults to the minimum field as defined in the start of authority (SOA) record for that zone. TTL is used when data is stored outside of a zone (in a cache) to ensure that the data does not stay around indefinitely.

For detailed information on how to set up specific types of name servers, see the following:

- How to Configure a Primary Name Server
- How to Configure a Secondary Name Server
- How to Configure a Cache-Only Name Server.

For further information on naming, read the DOMAIN Protocol on page 14–71.

Related Information

The **named** daemon.

The **named.boot** file, **named Cache** file, **named Local** file, **named Data** file, **named Reverse Data** file **resolv.conf** file, **hosts** file.

Planning for DOMAIN Name Resolution on page 14–108, How to Configure a Primary Name Server on page 14–109, How to Configure a Secondary Name Server on page 14–112, How to Configure a Cache-Only Name Server on page 14–114, How to Configure a Host to Use a Name Server on page 14–116.

The Standard Resource Record Format.

Understanding Addresses for TCP/IP

TCP/IP includes an Internet addressing scheme that allows users and applications to identify a specific network or host with which to communicate. An Internet address works like a postal address, allowing data to be routed to the chosen destination. TCP/IP provides standards for assigning addresses to networks, sub-networks, hosts, and sockets, and for using special addresses for broadcasts and local loopback.

Internet addresses are made up of a *network address* and a *local address.* This two part address allows a sender to specify the network as well as a specific host on the network. A unique, official network address is assigned to each network when it connects to other Internet networks. However, if a local network is not going to connect to other Internet networks, it can be assigned any network address that is convenient for local use.

The Internet addressing scheme consists of Internet (IP) addresses and two special cases of IP address: broadcast addresses and loopback addresses.

For further information on addresses, read Internet Addresses for TCP/IP on page 14–31, Subnet Addresses for TCP/IP on page 14–33, Broadcast Addresses for TCP/IP on page 14–36, and Local Loopback Addresses for TCP/IP on page 14–36.

For further information on addresses and naming, read Understanding Naming for TCP/IP on page 14–24.

A Class C address consists of a 24-bit network address and an 8-bit local host address. The first two bits in the network address are dedicated to indicating the network class, leaving 22 bits for the actual network address. Therefore there are 2,097,152 possible network addresses and 256 possible local host addresses. In a Class C address, the highest order bits are set to 1 and 1 as shown in the following diagram.

Class C Address

Network Address (24 bits)			**Local Host Address** (8 bits)
11011101	00001101	01001001	00001111

Note: The two highest order bits (or first two bits) will always be one and one in a Class C Address.

When deciding what class of network address to use, the person who administers your system needs to consider how many local hosts there will be on the network, and how many subnetworks there will be in the organization. If the organization is small and the network will have fewer than 256 hosts, a Class C address is probably sufficient. If the organization is large, then a Class B or Class A address may be more appropriate.

Note: Other systems may support Class D addresses, which are multicast addresses with the highest order bits set to 1-1-1. TCP/IP does not support Class D addresses.

Machines read addresses in binary code. The conventional notation for Internet host addresses is the *dotted decimal*, which divides the 32-bit address into four 8-bit fields. The following binary value:

```
0001010    00000010    00000000    00110100
```

can be expressed as

```
010.002.000.052      or    10.2.0.52
```

where the value of each field is specified as a decimal number and the fields are separated by periods.

TCP/IP requires a unique Internet address for each network interface (adapter) on a network. These addresses are determined by entries in the configuration database, which must agree with entries in the **/etc/hosts** file or the **named** database if the network is using a name server.

For further information on addresses, read Understanding Subnet Addresses for TCP/IP on page 14–33, Understanding Broadcast Addresses for TCP/IP on page 14–36, and Understanding Local Loopback Addresses for TCP/IP on page 14–36.

Related Information

The **hosts** file.

If the width of the *subnet_address* field is 0, the network is not organized into subnets, and addressing to the network is performed using the Internet network address.

The bits that identify the subnet are specified by a bit *mask* and, therefore, are not required to be adjacent in the address. However, it is generally desirable for the subnet bits to be contiguous and located as the most significant bits of the local address.

Subnet Masks

When a host sends a message to a destination, the system must determine whether the destination is on the same network as the source or if the destination can be reached directly through one of the local interfaces. The system compares the destination address to the host address using the *subnet mask*. If the destination is determined not to be local, the system sends the message on to a gateway. The gateway performs the same comparison to see if the destination address is on a network it can reach locally.

The subnet mask tells the system what the subnet partitioning scheme is. This bit mask consists of the Network Address portion and Subnet Address portion of the Internet address. For example, the subnet mask of the Class A address with the partitioning scheme defined above is shown in the following diagram:

Class A Address with Corresponding Subnet Address

Network Address (8 bits)	Local Host Address (24 bits)	
Network Address	**Subnet Address**	**Host Address**
01111101	00001101 0100	1001 00001111

Class A Address with Corresponding Subnet Mask

Network Address (8 bits)	Local Host Address (24 bits)	
Network Address	Subnet Address	Host Address
Subnet Mask		**Host Address**
01111101	00001101 0100	1001 00001111

The subnet mask is a set of 4 bytes, just like the internetwork address. The subnet mask consists of high bits (1s) corresponding to the bit positions of the network and subnetwork address, and low bits (0s) corresponding to the bit positions of the host address. A subnet mask for the above address looks like the following diagram:

Example Subnet Mask

Network Address (8 bits)	Local Host Address (24 bits)	
Network Address	**Subnet Address**	**Host Address**
11111111	11111111 1111	0000 00000000

The destination address and the local network address are compared by performing the logical AND and exclusive OR on the subnet mask of the source host.

Understanding Broadcast Addresses for TCP/IP

The TCP/IP can send data to all hosts on a local network, or to all hosts on all directly connected networks. Such transmissions are called *broadcast messages*. For example, the **routed** routing daemon uses broadcast messages to query and respond to routing queries.

For data to be broadcast to all hosts on all directly connected networks, User Datagram Protocol and Internet Protocol are used to send the data, and the host destination address in the IP header has all bits set to 1 (one). For data to be broadcast to all hosts on a specific network, all bits in the local address part of the IP address are set to 0. There are no user commands that use the broadcast capability, although such commands, or programs, can be developed.

The broadcast address can be changed temporarily by changing the *broadcast* parameter in the **ifconfig** command. Change the broadcast address permanently by using SMIT. Changing the broadcast address may be useful if you need to be compatible with older versions of software that used a different broadcast address; for example, the host IDs are all set to 0.

For further information on addresses, read Understanding Internet Addresses for TCP/IP on page 14–31, Understanding Subnet Addresses for TCP/IP on page 14–33, and Understanding Local Loopback Addresses for TCP/IP on page 14–36.

Related Information

The **ifconfig** command.

The **gated** daemon, **routed** daemon.

Understanding the User Datagram Protocol (UDP) on page 14–66.

Understanding Local Loopback Addresses for TCP/IP

The Internet Protocol defines the special network address, `127.0.0.1`, as a local loopback address. Hosts use local loopback addresses to send messages to themselves. The local loopback address is set by the configuration manager during the system startup process. Local loopback is implemented in the kernel and can also be set with the **ifconfig** command. Loopback will be invoked when the system is started.

For further information on addresses, read Understanding Internet Addresses for TCP/IP on page 14–31, Understanding Subnet Addresses for TCP/IP on page 14–33, and Understanding Broadcast Addresses for TCP/IP on page 14–36.

Related Information

The **ifconfig** command.

Use the **gated** daemon when you need to use the RIP, EGP, and DCN (HELLO) protocols on the same host or gateway. The **routed** daemon supports the RIP gateway protocol only.

Note: Unpredictable results can occur when the **gated** and **routed** daemons run on the same host.

For further information on routing, read Understanding Static and Dynamic Routing for TCP/IP on page 14–38, Understanding Gateways for TCP/IP on page 14–39, and Understanding Addressing for TCP/IP on page 14–30.

For further information on routing and protocols, read Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the HELLO Protocol (HELLO) on page 14–77, and Understanding the Routing Information Protocol (RIP) on page 14–78.

Related Information

The **route** command, **netstat** command.

The **gated** daemon, **routed** daemon.

How to Configure the routed Daemon on page 14–41, How to Configure the gated Daemon on page 14–103.

Understanding Static and Dynamic Routing for TCP/IP

In TCP/IP, routing can be one of two types: *static routing*, which uses manual input to update the routing table, and *dynamic routing*, which uses daemons to update the routing table automatically when new information is received.

Static routing can be practical for a single network communicating with one or two other networks. However, as your network begins to communicate with more networks, the number of gateways increases, and so does the amount of time and effort required to update the routing table manually.

Dynamic routing uses routing daemons (the **routed** daemon and the **gated** daemon) in two ways. In *active* mode, the daemon both broadcasts its own routing information and automatically updates the routing table as new information on routing is received. In *passive* mode, the daemon does not broadcast routing information, but waits to receive messages about routing information from other gateways and then updates the routing table.

Dynamic routing using daemons increases the level of activity on a network system. You may choose to use a combination of static and dynamic routing on your system.

These two types of routing can be used not only for gateways, but for other hosts on a network, as well. Static routing works the same for gateways as for other hosts. Dynamic routing daemons, however, must be run in the passive (quiet) mode when run on a host that is not a gateway. That is, the daemon only receives information from other routing sources and does not broadcast its own routing information.

For further information on routing, read Understanding Gateways for TCP/IP on page 14–39 and Understanding Addressing for TCP/IP on page 14–30.

For further information on routing and protocols, read Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the HELLO Protocol (HELLO) on page 14–77, Understanding the Routing Information Protocol (RIP) on page 14–78, and Simple Network Management Protocol (SNMP) on page NO TAG.

The routing information is sent in a pair, (N,D), where N is a network and D is a distance reflecting the cost of reaching the specified network. Each gateway advertises the networks it can reach and the costs of reaching them. The receiving gateway calculates the shortest paths to other networks and passes this information along to its neighbors. Thus, each exterior gateway is continually receiving routing information, updating its routing table and then passing that information to its exterior neighbors.

Gateway Protocols

All gateways, whether interior or exterior, use protocols to communicate with each other. Short descriptions of the TCP/IP gateway protocols follow:

HELLO Protocol (HELLO)

HELLO is one of the two protocols the interior gateways use to communicate among themselves. HELLO calculates the shortest path to other networks using delay time.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) is one of the two protocols the interior gateways use to communicate among themselves. Like the HELLO Protocol, RIP calculates the shortest path to other networks. Unlike HELLO, RIP estimates distance not by delay time, but by hop counts. The **gated** daemon stores all metrics internally as time delays, and converts all RIP hop counts to time delays.

Exterior Gateway Protocol (EGP)

The exterior gateways use the Exterior Gateway Protocol (EGP) to communicate among themselves. The EGP does not calculate the shortest path to other networks.

For further information on gateways, read Configuring Gateways for TCP/IP on page 14–41, Understanding Routing for TCP/IP on page 14–37, and Understanding Protocols for TCP/IP on page 14–54.

For further information on gateway protocols, read Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the HELLO Protocol (HELLO) on page 14–77, Understanding the Routing Information Protocol (RIP) on page 14–78, and Simple Network Management Protocol (SNMP) on page NO TAG.

Related Information

The **routed** daemon, **gated** daemon.

How to Configure the routed Daemon on page 14–107, How to Configure the gated Daemon on page 14–103.

If Using Dynamic Routing

Dynamic routing is a good idea for larger networks or larger autonomous systems. You might decide to use dynamic routing because you have a large autonomous system with several gateways and many hosts. If you have different types of networks, you probably also have different types of protocols. The daemon you choose is important. Choose the routing daemon you use according to the type of gateway you need and the protocols your gateway is required to support.

Consider the Protocols to Use

Interior gateways use the HELLO Protocol (HELLO) and Routing Information Protocol (RIP). Exterior gateways use the Exterior Gateway Protocol (EGP).

Choose a Daemon to Use

The two routing daemons in TCP/IP are the **routed** and **gated** daemons. The two daemons *cannot* be run together on the same gateway.

The **gated** daemon supports the Exterior Gateway Protocol (EGP), the HELLO Protocol (HELLO), the Routing Information Protocol (RIP), and the Simple Network Monitoring Protocol (SNMP). By using the **gated** daemon your gateway can communicate with interior gateways and with exterior gateways. If you use the EGP, you should obtain an *autonomous system number* from Internet for your gateway. Also, find out who your EGP neighbors are.

Note: SNMP is not a routing protocol; it is used to change or show management information for a network element from a remote host.

Through System Resource Control (SRC), the **gated** daemon can run in active or passive mode, trace packets sent and received, and log debugging information.

The **routed** daemon supports the Routing Information Protocol (RIP) only. By using the **routed** daemon your gateway can communicate with other interior gateways within the autonomous system, but not with exterior gateways outside the autonomous system. Through SRC, the **routed** daemon can run in active or passive mode, trace packets sent and received, and log debugging information.

Configure the Daemon

Set up the daemons by altering the appropriate configuration files.

For further information on gateways, read Understanding Routing for TCP/IP on page 14–37, and Understanding Protocols for TCP/IP on page 14–54, and Understanding the TCP/IP Daemons on page 14–19.

For further information on gateway protocols, read Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the HELLO Protocol (HELLO) on page 14–77, and Understanding the Routing Information Protocol (RIP) on page 14–78.

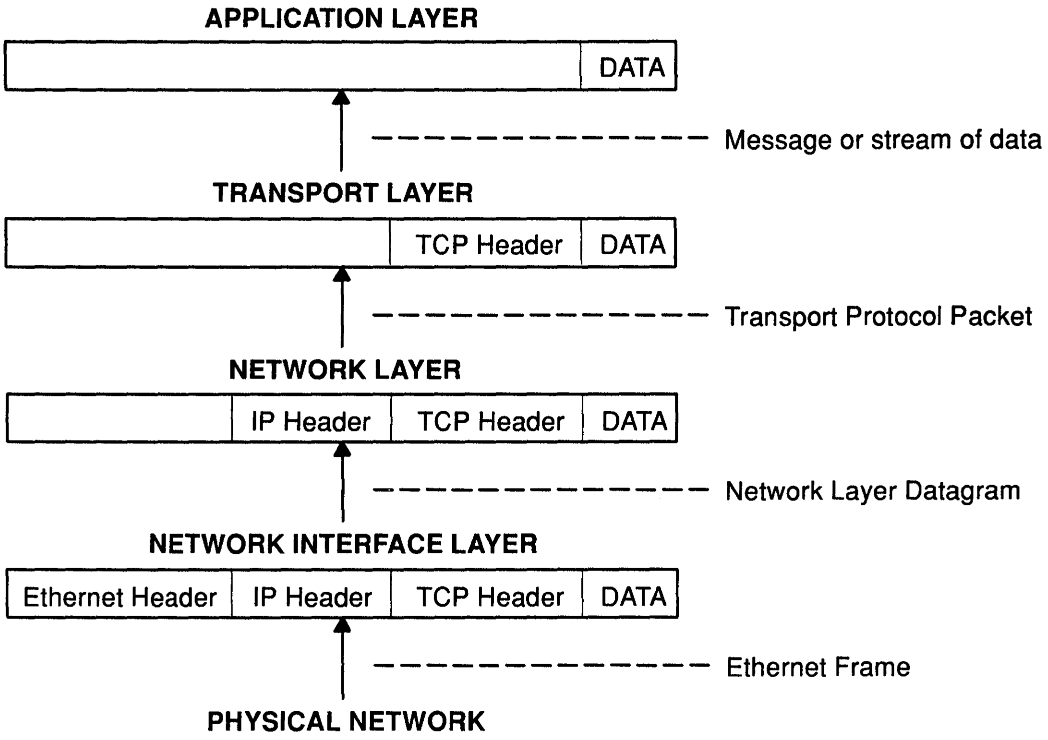
Related Information

The **gated** daemon, **routed** daemon.

The **gated.conf** file format, **/etc/gateways** file format, **/etc/networks** file format.

How to Configure the routed Daemon on page 14–107, How to Configure the gated Daemon on page 14–103.

Destination Host



Packet headers are stripped by the destination host.

Packet Tracing

Packet tracing is the process by which you can verify the path of a packet through the layers to its destination. The **iptrace** command performs network interface level packet tracing. The **ipreport** command issues output on the packet trace in both hexadecimal and ASCII format. The **trpt** command performs transport protocol level packet tracking for the Transmission Control Protocol (TCP). The **trpt** command output is more detailed, including information on time, TCP state, and packet sequencing.

For further information on packets and protocols, read Address Resolution Protocol (ARP) on page 14–58, Internet Protocol (IP) on page 14–61, Transmission Control Protocol (TCP) on page 14–67, and User Datagram Protocol (UDP) on page 14–66.

For further information on packets and routing, read Understanding Routing for TCP/IP on page 14–37.

For further information on packets and network interfaces, read Understanding Network Interfaces for TCP/IP on page 14–45 and Understanding Network Interface Packet Headers on page 14–47.

Related Information

The **iptrace** command, **ipreport** command, **trpt** command.

How to Configure the routed Daemon on page 14–107, How to Configure the gated Daemon on page 14–103.

TCP/IP supports six types of network interfaces, as follows:

- Ethernet (en)
- 802.3 (et)
- Token-Ring (tr)
- X.25 Protocol (xt)
- Serial Line Internet Protocol, or SLIP (sl)
- Loopback (lo).

The Ethernet, 802.3, and Token-Ring interfaces are for use with local area networks (LANs). The SLIP and X.25 interfaces are for use with serial connections. The loopback interface is used by a host to send messages back to itself.

Configuring a Network Interface

Network interfaces are configured automatically at system startup. The configuration uses default values. To customize a network interface use the System Management Interface Tool (SMIT).

Note: In network adapter configuration, the Receive Data Transfer Offset (RDTO) field indicates where in the receive buffer the packet data actually begins. To optimize data transfer performance, during network adapter configuration set the value for the device characteristic RDTO to the following:

ethernet	26
token–ring	0
802.3	18
X.25	12

For information on how to set RDTO, see the description of defining an Ethernet Adapter, Token-Ring Adapter, Multiprotocol Adapter, or X.25 Adapter.

For further information on network interfaces, read Network Adapter Cards for TCP/IP on page 14–50, Automatic Configuration of Network Interfaces for TCP/IP on page 14–51, Understanding Packets for TCP/IP on page 14–43, and Understanding Gateways for TCP/IP on page 14–39.

For further information on configuring a network interface, read SMIT Interface for TCP/IP on page 14–88 and Understanding Installation and Configuration for TCP/IP on page 14–80.

Related Information

How to Configure a Network Interface for TCP/IP on page 14–100.

Understanding the SMIT Interface for TCP/IP on page 14–88.

Token-Ring and 802.3 Local Headers

The following represents an IP or ARP local header for the Token-Ring Adapter or the 802.3 Adapter:

Medium Access Control (MAC) Header Logical Link Control (LLC) Header

The Medium Access Control (MAC) header for the Token-Ring Adapter is composed of five fields, as shown below:

MAC Header		
Field	**Length**	**Definition**
AC	1 byte	Access Control. The value in this field x'00' gives the header priority 0.
FC	1 byte	Field Control. The value in this field x'40' specifies the Logical Link Control frame.
DA	6 bytes	Destination Address.
SA	6 bytes	Source Address. If bit 0 of this field is set to 1, it indicates that routing information (RI) is present.
RI	<18 bytes>	Routing Information. The valid fields are discussed below.

Routing Information Fields for the MAC Header

Field	**Byte Space**
RC	2 bytes
Segment Numbers (up to 8)	2 bytes each.

The RI field definitions follow:

RC
Routing Control. RC information is contained in bytes 0 and 1 of the RI field. The settings of the first two bits of the RC field have the following meanings:

bit (0) = 0 Use the non-broadcast route specified in the RI field.

bit (0) = 1 Create the RI field and broadcast to all rings.

bit (1) = 0 Broadcast through all bridges.

bit (1) = 1 Broadcast through limited bridges.

Segment Numbers
Up to eight segment numbers of 2 bytes each to specify recipients of a limited broadcast.

Understanding Network Adapter Cards for TCP/IP

The network adapter card is the hardware that is physically attached to the network cabling. It is responsible for receiving and transmitting data at the physical level. The network adapter card is controlled by the network adapter device driver.

A machine must have one network adapter card (or connection) for each network (not network type) to which it connects. For instance, if a host attaches to two token-ring networks, it must have two network adapter cards.

TCP/IP uses the following network adapter cards and connections:

- Ethernet/802.3
- Token-Ring
- Asynchronous adapters, native serial ports
- X.25.

The Ethernet and 802.3 network technologies use the same type of adapter.

TCP/IP has a maximum number of adapters it can support on any one machine. It will support up to four Ethernet/802.3 adapters, four token-ring adapters, four X.25 adapters, and one asynchronous adapter card with up to eight connections.

Note: The maximum number of a type of adapter on a host is software-dependent. The number of adapters you can install on any one machine also depends on the number of slots the machine has available.

Automatic Configuration

The system startup process is able to determine which adapters exist within the system. At each system startup, the network interface software will automatically configure using default values. You can change the default values by using the System Management Interface Tool (SMIT). The machine will also load the appropriate network adapter device driver.

For further information on configuring a network interface, read Understanding the SMIT Interface for TCP/IP on page 14–88 and Understanding Configuration for TCP/IP on page 14–80.

Related Information

How to Configure a Network Interface on page 14–100.

The following is a list of valid 802.3 network device driver attributes along with their default values as shown under the Network Interface Driver menu in SMIT.

ATTRIBUTE	DEFAULT VALUE	POSSIBLE VALUES
mtu	1492	60 through 1492
remmtu	576	60 through 1492

Token-Ring Default Configuration Values

The following is a list of valid Token-Ring network adapter attributes along with their default values as shown under the Network Interface Selection menu in SMIT.

ATTRIBUTE	DEFAULT VALUE	POSSIBLE VALUES
netaddr		
state	down	up,down
trailers	off	on,off
arp	on	on,off
allcast	on	on,off
hwloop	off	on,off
netmask		
broadcast		
security	none	unclassified, confidential secret, top_secret, none
authorized		genser, nsiop, ndsccs–spintcom, ndsccs–critcom

The following is a list of valid Token-Ring network device driver attributes along with their default values as shown under the Network Interface Driver menu in SMIT.

ATTRIBUTE	DEFAULT VALUE	POSSIBLE VALUES
mtu	1500	60 through 4096
remmtu	576	60 through 4096

Note: The upper limit of the possible values may be increased in the future to accommodate the new high bandwidth token-ring network (16MB token-ring network or FDDI).

X.25 Default Configuration Values

The following is a list of valid X.25 network adapter attributes along with their default values as shown under the Network Interface Selection menu in SMIT.

ATTRIBUTE	DEFAULT VALUE	POSSIBLE VALUES
netaddr		
state	down	up,down
netmask		

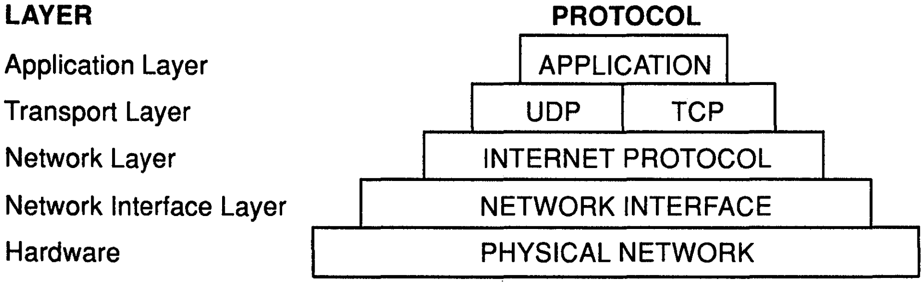
The following is a list of valid X.25 network device driver attributes along with their default values as shown under the Network Interface Driver menu in SMIT.

ATTRIBUTE	DEFAULT VALUE	POSSIBLE VALUES
mtu	576	60 through 1024
remmtu	576	60 through 4096

Understanding Protocols for TCP/IP

Protocols are sets of rules for message formats which allow machines and application programs to send information to each other. These rules must be followed by each machine involved in the communication in order for the receiving host to be able to understand the message. In order to transmit the information, the sending host must break it down into small pieces called *packets*. The receiving host then reassembles the packets into the original message.

The TCP/IP *suite* of protocols can be understood in terms of layers, as in the following diagram.



TCP/IP Suite of Protocols

On a TCP/IP network, an authorized user can log in remotely to a machine while remaining at another workstation. If the user types in a single letter at the local keyboard, the letter is passed through the layers of protocols, which reformat it into machine-readable bits of information that are then sent across the network. At the lowest level, a protocol readies the information for the hardware to carry the datagram to the destination machine. The destination host receives the datagram and sends it back up through the layers of protocols, which reassemble the datagram into the letter originally typed by the user.

Applications programs send messages or streams of data to the top layer, Internet Transport Level Protocols, which include the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP). These protocols receive the data from the application, divide it into small pieces called *packets*, add a destination address, and then pass the packets along to the next protocol layer, the Internet Protocol.

The Internet Protocol layer encloses the packet in an IP datagram, puts in the datagram header and trailer, decides where to send the datagram (either directly to a destination or else to a gateway), and passes the datagram on to the Network Interface layer. The Network Interface protocol accepts IP datagrams and transmits them as frames over a specific network hardware (such as Ethernet or Token-Ring networks).

Frames received by a network go through the protocol layers in reverse. Each layer strips off the corresponding header information, until the data is back at the application level. Frames are received by the Network Interface Layer (in this case, an Ethernet adapter). The Network Interface Layer strips off the Ethernet header, and sends the datagram up to the Network layer. In the Network Layer, the Internet Protocol strips off the IP header and sends the packet up to the Transport Layer. In the Transport Layer, the Transmission Control Protocol (in this case) strips off the TCP header and sends the data up to the Application Layer.

Application-Level Internet Protocols

- Domain Name Protocol (DOMAIN)
- Exterior Gateway Protocol (EGP)
- File Transfer Protocol (FTP)
- Name/Finger Protocol (FINGER)
- Telnet Protocol (TELNET)
- Trivial File Transfer Protocol (TFTP).

Application-Level Protocols

- DCN Local-Network Protocol (HELLO)
- Remote Command Execution Protocol (EXEC)
- Remote Login Protocol (LOGIN)
- Remote Shell Protocol (SHELL)
- Routing Information Protocol (RIP)
- Time Server Protocol (TIMED).

For further information on TCP/IP protocols, read Understanding Internet Network-Level Protocols on page 14–57, Understanding Internet Transport–Level Protocols on page 14–65, Understanding Internet Application–Level Protocols on page 14–70, Understanding TCP/IP Application–Level Protocols on page 14–77, and Understanding the X.25 Protocol on page 14–79.

For further information on TCP/IP communication, read Understanding Addresses for TCP/IP on page 14–30, Understanding Packets for TCP/IP on page 14–43, and Understanding Routing for TCP/IP on page 14–37.

Related Information

The **lpd** daemon.

Understanding the Address Resolution Protocol (ARP)

The Address Resolution Protocol (ARP) translates Internet addresses, which are unique, into the unique hardware addresses for the following adapters:

- Ethernet LAN Adapter (supports both Ethernet and 802.3 protocols)
- IBM Token-Ring Network Adapter.

ARP does not translate addresses for X.25 or SLIP since these are point-to-point connections.

ARP dynamically maps Internet addresses to hardware addresses on local area networks and stores the information in mapping tables. TCP/IP uses ARP to collect and distribute the information for address mapping.

The kernel maintains the translation tables, and the ARP is not directly available to users or applications. When an application sends an Internet packet to one of the interface drivers, the driver requests the appropriate address mapping. If the mapping is not in the table, an ARP broadcast packet is sent through the requesting interface driver to the hosts on the local area network.

Entries in the ARP mapping table are deleted after 20 minutes; incomplete entries are deleted after 3 minutes. To make a permanent entry in the ARP mapping tables, use the **arp** command with the *pub* parameter:

```
arp –s 802.3 host2 0 dd:0:a:8s:0 pub.
```

When any host that supports ARP receives an ARP request packet, the host notes the IP and hardware addresses of the requesting system and updates its mapping table, if necessary. If the receiving host IP address does not match the requested address, the host discards the request packet. If the IP address does match, the receiving host sends a response packet to the requesting system. The requesting system stores the new mapping and uses it to transmit any similar pending Internet packets.

Unlike most protocols, ARP packets do not have fixed-format headers.Instead, the message is designed to be useful with a variety of network technologies.

For further information on address resolution, read Understanding Addresses for TCP/IP on page 14–30.

For further information on Internet network-level protocols, read Internet Control Message Protocol (ICMP) on page 14–59, Internet Protocol (IP) on page 14–61, and VAX Trailer Encapsulation Protocol (VAX) on page 14–64.

Related Information

The **arp** command.

Understanding Internet Control Message Protocol Message Types

ICMP sends and receives the following message types:

echo request Sent by hosts and gateways to test whether a destination is alive and reachable.

information request Sent by hosts and gateways to obtain an Internet address for a network to which they are attached. This message type is sent with the network portion of IP destination address set to zero (0).

timestamp request Sent to request that the destination machine return its current value for time of day.

address mask request Sent by host to learn its subnet mask. The host can either send to a gateway, if it knows the gateway address, or send a broadcast message.

destination unreachable Sent when a gateway cannot deliver an IP datagram.

source quench Sent by discarding machine when datagrams arrive too quickly for a gateway or host to process, in order to request that the original source slow down its rate of sending datagrams.

redirect message Sent when a gateway detects that some host is using a non-optimum route.

echo reply Sent by any machine that receives an echo request in reply to the machine that sent the request.

information reply Sent by gateways in response to requests for network addresses, with both the source and destination fields of the IP datagram specified.

timestamp reply Sent with current value of time of day.

address mask reply Sent to machines requesting subnet masks.

parameter problem Sent when a host or gateway finds a problem with a datagram header.

time exceeded Sent when...

- Each IP datagram contains a time-to-live counter (hop count), which is decremented by each gateway.
- A gateway discards a datagram because its hop count has reached 0 (zero).

Internet Timestamp Used to record the time stamps through the route.

For further information, read Understanding Packets for TCP/IP on page 14–43.

Fragment Offset
Specifies the offset of this fragment in the original datagram measured in units of 8 octets.

Time To Live
Specifies how long the datagram can remain on the Internet. This keeps misrouted datagrams from remaining on the Internet forever.

Protocol
Specifies the high-level protocol type.

Header Checksum
Indicates a number computed to ensure the integrity of header values.

Source Address
Specifies the Internet address of the sending host.

Destination Address
Specifies the Internet address of the receiving host.

Options
Used mostly for network testing and debugging. This field is not required for every datagram.

- **End of Option List**
  Indicates the end of the option list. It is used at the end of all options, not at the end of each option individually, and need be used only if the end of the options would not otherwise coincide with the end of the IP header. End of Option List is used if options exceed the length of the datagram.

- **No Operation**
  Used between other options; for example, to align the beginning of a subsequent option on a 32-bit boundary.

- **Security**
  Provides a way for hosts to send security, compartmentation, handling restrictions, and TCC (closed user group parameters).

- **Loose Source and Record Route**
  Provides a means for the source of an Internet datagram to supply routing information to be used by the gateways in forwarding the datagram to a destination and for recording the route information.This is a *loose* source route: the gateway or host IP is allowed to use any route of any number of other intermediate gateways in order to reach the next address in the route.

- **Strict Source and Record Route**
  Provides a means for the source of an Internet datagram to supply routing information to be used by the gateways in forwarding the datagram to a destination and for recording the route information. This is a *strict* source route: the gateway or host IP must send the datagram directly to the next address in the source route through only the directly connected network indicated in the next

Understanding the VAX Trailer Encapsulation Protocol (VAX)

VAX Trailer Encapsulation Protocol supports VAX trailers. It moves all variable-length header information in an IP packet to a position following the data segment. Trailer encapsulation allows the receiving host to receive data on a page-aligned boundary, which is a requirement for utilizing a page-mapped virtual memory environment. The TCP/IP receives and processes VAX trailer protocol data and can be configured to send it, using the **ifconfig** command.

For further information on Internet network-level protocols, read Understanding the Address Resolution Protocol (ARP) on page 14–58, Understanding the Internet Control Message Protocol (ICMP) on page 14–59, and Understanding the Internet Protocol (IP) on page 14–61.

Related Information

The **ifconfig** command.

Understanding the User Datagram Protocol (UDP)

Sometimes an application on a network needs to send messages to a specific application or process on another network. UDP provides a datagram means of communication between applications on Internet hosts. Because senders do not know which processes are active at any given moment, UDP uses destination protocol ports (or abstract destination points within a machine), identified by positive integers, to send messages to one of multiple destinations on a host. The protocol ports receive and hold messages in queues until applications on the receiving network can retrieve them.

Since UDP relies on the underlying IP to send its datagrams, UDP provides the same connectionless message delivery as IP. It offers no assurance of datagram delivery or duplication protection. However, UDP does allow the sender to specify source and destination port numbers for the message and also calculates a checksum of both the data and header. These two features allow the sending and receiving applications to ensure the correct delivery of a message. The following diagram depicts a UDP header.

Bits

0	16 31
SOURCE PORT NUMBER	DESTINATION PORT NUMBER
LENGTH	CHECKSUM

User Datagram Protocol (UDP) Packet Header

UDP Header Field Definitions

Source Port Number The address of the protocol port sending the information.

Destination Port Number The address of the protocol port receiving the information.

Length The length in octets of the UDP datagram.

Checksum Provides a check on the UDP datagram using the same algorithm as the Internet Protocol (IP).

The API to UDP is a set of library subroutines provided by the sockets interface.

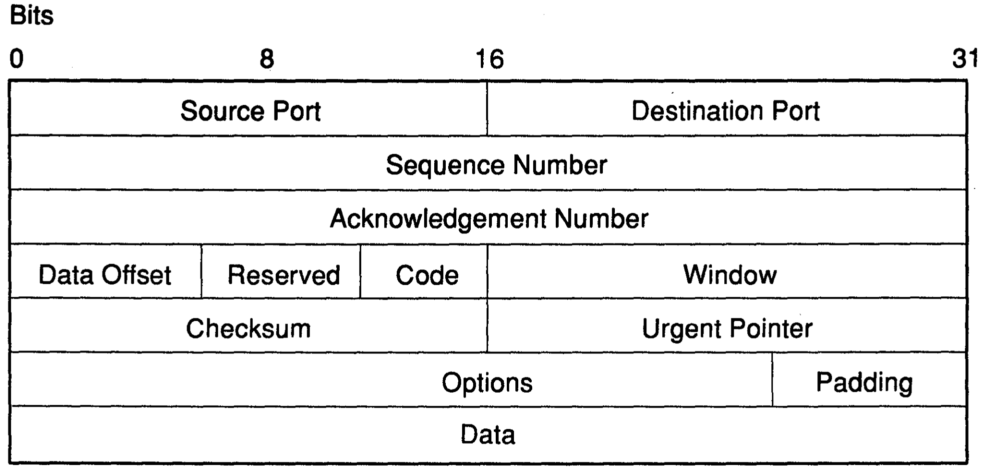
Note: Applications that require reliable delivery of datagrams must implement their own reliability checks when using UDP. Applications that require reliable delivery of streams of data should use TCP.

For further information on Internet transport-level protocols, read Understanding the Transmission Control Protocol (TCP) on page 14–67.

Related Information

Sockets Overview in *General Programming Concepts.*

The following is a diagram of the TCP packet header:



Transmission Control Protocol (TCP) Packet Header

TCP Header Field Definitions

Source Port Identifies the port number of a source application program.

Destination Port Identifies the port number of a destination application program.

Sequence Number Specifies the sequence number of the first byte of data in this segment.

Acknowledgement Number
Identifies the position of the highest byte received.

Data Offset Specifies the Offset of data portion of the segment.

Reserved Reserved for future use.

Code Control bits to identify the purpose of the segment:

URG	Urgent pointer field is valid.
ACK	Acknowledgement field is valid.
PSH	The segment requests a PUSH.
RTS	Resets the connection.
SYN	Synchronizes the sequence numbers.
FIN	Sender has reached the end of its byte stream.

Window Specifies the amount of data the destination is willing to accept.

Checksum Verifies the integrity of the segment header and data.

Urgent Pointer Indicates data that should be delivered as quickly as possible. This pointer specifies the position where urgent data ends.

Understanding Internet Application–Level Protocols

TCP/IP implements higher level Internet protocols at the application program level. When an application needs to send data to another application on another host, the applications send the information down to the Transport level protocols to prepare the information for transmission. These protocols include:

- Domain Name Protocol (DOMAIN)
- Exterior Gateway Protocol (EGP)
- File Transfer Protocol (FTP)
- Name/Finger Protocol (FINGER)
- Telnet Protocol (TELNET)
- Trivial File Transfer Protocol (TFTP).

The following diagram depicts the protocol hierarchy:

LAYER	PROTOCOL	
Application Layer	APPLICATION	
Transport Layer	UDP	TCP
Network Layer	INTERNET PROTOCOL	
Network Interface Layer	NETWORK INTERFACE	
Hardware	PHYSICAL NETWORK	

Application Layer of the TCP/IP Suite of Protocols

TCP/IP does not provide APIs to these protocols.

For further information on other types of protocols, read Understanding Internet Network-Level Protocols on page 14–57, Understanding Internet Transport-Level Protocols on page 14–65, Understanding TCP/IP Application-Level Protocols on page 14–77, and Understanding the X.25 Protocol on page 14–79.

For further information on Internet application-level protocols, read Understanding the Domain Name Protocol (DOMAIN) on page 14–71, Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the File Transfer Protocol (FTP) on page 14–74, Understanding the Name/Finger Protocol (FINGER) on page 14–76, Understanding the Telnet Protocol (TELNET) on page 14–75, and Understanding the Trivial File Transfer Protocol (TFTP) on page 14–76.

Understanding the Exterior Gateway Protocol (EGP)

Exterior Gateway Protocol (EGP) is the mechanism that allows the exterior gateway of an *autonomous system* to share routing information with exterior gateways on other autonomous systems.

An autonomous system is a group of networks and gateways for which one administrative authority has responsibility. Gateways are *interior neighbors* if they reside on the same autonomous system and *exterior neighbors* if they reside on different autonomous systems. Gateways that exchange routing information using EGP are said to be EGP *peers* or *neighbors*. Autonomous system gateways use EGP to provide reachability information to their EGP neighbors.

EGP allows an exterior gateway to ask another exterior gateway to agree to exchange reachability information, continually checks to ensure that its EGP neighbors are responding, and helps EGP neighbors to exchange reachability information by passing routing update messages.

EGP restricts exterior gateways by allowing them to advertise only those destination networks reachable entirely within that gateway's autonomous system. Thus, an exterior gateway using EGP passes along information to its EGP neighbors but does not advertise reachability information about its EGP neighbors outside the autonomous system.

EGP does not interpret any of the distance metrics that appear in routing update messages from other protocols. EGP uses the distance field to specify whether a path exists (a value of 255 means that the network is unreachable). The value cannot be used to compute the shorter of two routes unless those routes are both contained within a single autonomous system. For this reason, EGP cannot be used as a routing algorithm. As a result there will be only one path from the exterior gateway to any network.

In contrast to the Routing Information Protocol (RIP), which can be used within an autonomous system of Internet networks that dynamically reconfigure routes, EGP routes are predetermined in the **/etc/gated.conf** file. EGP assumes that IP is the underlying protocol.

EGP Message Types

Neighbor Acquisition Request

Used by exterior gateways to request to become neighbors of each other.

Neighbor Acquisition Reply

Used by exterior gateways to accept the request to become neighbors.

Neighbor Acquisition Refusal

Used by exterior gateways to deny the request to become neighbors. The refusal message includes reasons for refusal, such as `out of table space`.

Neighbor Cease

Used by exterior gateways to cease the neighbor relationship. The cease message includes reasons for ceasing, such as `going down`.

Neighbor Cease Acknowledgment

Used by exterior gateways to acknowledge the request to cease the neighbor relationship.

Understanding the File Transfer Protocol (FTP)

File Transfer Protocol (FTP) allows hosts to transfer data among dissimilar hosts as well as files between two foreign hosts indirectly. FTP provides for such tasks as listing remote directories, changing the current remote directory, creating and removing remote directories, and transferring multiple files in a single request. FTP keeps the transport secure by passing user and account passwords to the foreign host. Although FTP is designed primarily to be used by applications, it also allows interactive user-oriented sessions.

FTP uses reliable stream delivery (TCP/IP) to send the files and uses a TELNET connection to transfer commands and replies. FTP also understands several basic file formats including ASCII, IMAGE, and Local 8.

TCP/IP implements FTP in the **ftp** user command and the **ftpd** server command and does not provide an API to this protocol.

AIX implements user-server and server-server relationships through the **ftp** command. The **ftp** command provides the server with the following configuration:

TYPE	ASCII Binary Form non_print
MODE	Stream
STRU	File
COMM	MODE, NOOP, PORT, QUIT, RETR, STOR, STRU, TYPE, USER

Default values for transfer parameters are:

TYPE	ASCII Form non_print
MODE	Stream
STRU	File

For further information on Internet application-level protocols, read Understanding the Domain Name Protocol (DOMAIN) on page 14–71, Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the Name/Finger Protocol (FINGER) on page 14–76, Understanding the Telnet Protocol (TELNET) on page 14–75, and Understanding the Trivial File Transfer Protocol (TFTP) on page 14–76.

Related Information

The **ftp** command, **ftpd** daemon.

Understanding the Trivial File Transfer Protocol (TFTP)

The Trivial File Transfer Protocol (TFTP) can read and write files to and from a foreign host. Since TFTP uses the unreliable User Datagram Protocol (UDP) to transport files, it is generally quicker than FTP. Like FTP, TFTP can transfer files as either NETASCII characters or as 8-bit binary data. Unlike FTP, TFTP cannot be used to list or change directories at a foreign host and it has no provisions for security like password protection. And, data can be written or retrieved only in public directories.

The TCP/IP implements TFTP in the **tftp** and **utftp** user commands and in the **tftpd** server command. The **utftp** command is a form of the **tftp** command for use in a pipe. The TCP/IP does not provide an API to this protocol.

For further information on Internet application-level protocols, read Understanding the Domain Name Protocol (DOMAIN) on page 14–71, Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the File Transfer Protocol (FTP) on page 14–74, Understanding the Name/Finger Protocol (FINGER) on page 14–76, and Understanding the Telnet Protocol (TELNET) on page 14–75.

Related Information

The **tftp** command, **tftpd** daemon, **utftp** command.

Understanding the Name/Finger Protocol (FINGER)

The Name/Finger Protocol (FINGER) is an application-level Internet protocol that provides an interface between the **finger** command and the **fingerd** daemon. The **fingerd** daemon returns information about the users currently logged in to a specified remote host. If you execute the finger command specifying a user at a particular host, you will obtain specific information about that user. The Finger Protocol must be present at the remote host and at the requesting host. FINGER uses Transmission Control Protocol (TCP) as its underlying protocol.

The TCP/IP does not provide an API to this protocol.

For further information on Internet application-level protocols, read Understanding the Domain Name Protocol (DOMAIN) on page 14–71, Understanding the Exterior Gateway Protocol (EGP) on page 14–72, Understanding the File Transfer Protocol (FTP) on page 14–74, Understanding the Telnet Protocol (TELNET) on page 14–75, and Understanding the Trivial File Transfer Protocol (TFTP) on page 14–76.

Related Information

The **finger** command.

The **fingerd** daemon.

Remote Shell Protocol (SHELL)

The **rsh** user command and the **rshd** daemon provide the remote command shell protocol, which allows users to open a shell on a compatible foreign host for executing commands.

Routing Information Protocol (RIP)

Routing Information Protocol (RIP) and the **routed** and **gated** daemons that implement it keep track of routing information based on gateway hops and maintain kernel-routing table entries.

Time Server Protocol (TIMED)

The **timed** daemon is used to synchronize a host's time with the time of other hosts. It is based on the client/server concept.

For further information, read Understanding Internet Network-Level Protocols on page 14–57, Understanding Internet Transport-Level Protocols on page 14–65, Understanding Internet Application-Level Protocols on page 14–70, and Understanding the X.25 Protocol on page 14–79.

Related Information

The **gated** daemon, **lpd** daemon, **rexecd** daemon, **rlogind** daemon, **routed** daemon, **rshd** daemon, **timed** daemon.

The **rexec** command, **rlogin** command, **rsh** command.

Understanding Installation and Configuration for TCP/IP

Once the TCP/IP software is loaded on your system, you are ready to begin configuring the network. Since TCP/IP is a flexible networking tool, there are many ways you can customize TCP/IP to fit the specific needs of your organization.

Most configuration tasks can be done using the System Management Interface Tool (SMIT). Many of the configuration tasks can be done either through SMIT, by editing a file format, or by issuing a command. A few tasks, such as configuring a name server, cannot be done using SMIT.

Installing TCP/IP

TCP/IP software can be installed in several ways. Since TCP/IP is part of the Networks package, it can be installed along with other applications such as NFS. It can also be installed along with the Base Operating System (BOS).

TCP/IP can also be installed using the **installp** command.

```
installp –F bosnet.tcpip.obj
```

Once the software is loaded, you can configure TCP/IP to suit your needs.

For further information on TCP/IP installation, see How to Install AIX for RISC System/6000 Licensed Program Optional Program Products.

Configuring TCP/IP

Many TCP/IP configuration tasks can be performed in more than one way. In the AIX system, there is a tool called the System Management Interface Tool, or *SMIT*, which provides a menu driven means of performing many configuration tasks. Refer to the SMIT Interface for TCP/IP for a list of tasks in TCP/IP that can be performed using SMIT.

The **rc.net** shell script performs required host configuration for TCP/IP. The **rc.net** file can be modified using a standard editor to call traditional TCP/IP configuration commands such as **ifconfig**, **hostname**, and **route**. It runs automatically during the startup process.

The **rc.net** script is run by the configuration manager program during the second phase of configuration.

AIX TCP/IP also provides the traditional UNIX means of configuring TCP/IP. Refer to the List of TCP/IP System Management Commands, the List of TCP/IP File Formats, the List of TCP/IP Daemons, and the List of TCP/IP Procedures for further information.

Configuring Hosts

Each host machine on your network will need to be configured to function according to the needs of the end-users and the network as a whole. For each host on the network, you are required to configure the network interface, set the Internet address, and set the host name. To perform minimal configuration on a host, use the `smit startup` command, or the `smit mktcpip` command. See How To Configure a Host on a TCP/IP Network and the **mktcpip** command for more information.

You may also want to set up static routes to gateways or other hosts, specify daemons to be started by default, set up the **/etc/hosts** file for name resolution, or set up the host to use a name server for name resolution. If the host machine is to have a specific function, for example, if it is to serve as a gateway, file server, or name server, perform the necessary configuration tasks after the basic configuration is complete.

Related Information

The **routed** daemon, **gated** daemon, **named** daemon, **lpd** daemon.

The **hosts** file, **rc.net** file, **rc.tcpip** file.

Checklist for Configuring a TCP/IP Network on page 14–96, Planning Your TCP/IP Network on page 14–108.

How to Configure a Host on a TCP/IP Network on page 14–97, How to Configure a Network Interface on page 14–100, How to Configure a Primary Name Server on page 14–109, How to Configure a Secondary Name Server on page 14–112, How to Configure the gated Daemon on page 14–103, How to Configure the inetd Daemon on page 14–102, How to Configure the routed Daemon on page 14–107, Network Information Service Overview on page 11–47.

mktcpip Sets the required values for starting TCP/IP on a host.

namerslv Directly manipulates domain name server entries for local resolver routines in the system configuration database.

netstat Shows network status.

no Configures network options.

notinet Removes the **inetd** shared memory segment when the **inetd** daemon dies abnormally.

rmnamsv Unconfigures TCP/IP-based name service on a host.

rmprtsv Unconfigures a print service on a client or server machine.

route Manually manipulates the routing tables.

ruser Directly manipulates entries in three separate system databases that control foreign host access to programs.

securetcpip Enables the AIX network security feature.

setclock Sets the time and date for a host on a network.

slattach Attaches serial lines as network interfaces.

stinet Returns long status for **inetd** subservers.

stpinet Disables the inet instance.

sttinet Enables the inet instance.

timedc Returns information about the **timed** daemon.

trpt Performs protocol tracing on Transmission Control Protocol (TCP) sockets.

ucfgif Unloads an Interface instance from the kernel.

ucfginet Unloads the Internet instance and all related interface instances from the kernel.

udefif Removes an interface object from the system configuration database.

udefinet Undefines the Internet instance in the configuration database.

x25xlate Updates or displays translate information in the IP/X.25 translate table.

List of TCP/IP System Management File Formats

Domain Cache Defines the root name server or servers for a DOMAIN name server host.

Domain Data Stores name resolution information for the **named** daemon.

Domain Local Data Defines the local loopback information for **named** on the name server host.

Domain Reverse Data Stores reverse name resolution information for the **named** daemon.

ftpusers Specifies local user names that cannot be used by remote FTP clients.

gated.conf Contains configuration information for the **gated** daemon.

gateways Specifies Internet routing information to the **routed** and **gated** daemons on a network.

hosts Defines the Internet Protocol (IP) name and address of the local host and specifies the names and addresses of remote hosts.

hosts.equiv Specifies remote systems that can execute commands on the local system.

hosts.lpd Specifies remote hosts that can print on the local host.

inetd.conf Defines how the **inetd** daemon handles Internet service requests.

named.boot Defines how **named** initializes the DOMAIN name server file.

.netrc Specifies automatic login information for the **ftp** and **rexec** commands.

networks Contains the network name file.

protocols Defines the Internet protocols used on the local host.

rc.net Defines host configuration for the following areas: network interfaces, host name, default gateway, and any static routes.

rc.tcpip Initializes daemons each system IPL.

resolv.conf Defines DOMAIN name server information for local resolver routines.

.rhosts Specifies remote users that can use a local user account on a network.

services Defines the sockets and protocols used for Internet services.

Standard Resource Record Format
Defines the format of lines in the DOMAIN data files.

.3270keys Defines a user keyboard mapping and colors for TELNET (3270).

Understanding the SMIT Interface for TCP/IP

AIX provides a tool with which the person who manages your system can perform some common system management tasks. This tool is a set of dialogs called the *System Management Interface Tool*, or SMIT.

SMIT consists of a tree-structured set of menus attached to dialogs that prompt users for information needed to perform the task. The dialogs then call the command or process that implements it. Using the menu trees, you can begin with a general menu and choose options on each successive menu to get to the specific task you want to perform.

SMIT also provides a fast path to each menu and task that the tool provides. To use the fast path, type the **smit** command, followed by the keyword, as follows.

```
smit keyword
```

List of SMIT Fast Paths for TCP/IP

The following is a list of TCP/IP system management menus and tasks provided by the SMIT interface. The SMIT fast paths, or keywords, are shown in the parentheses following the items:

- Host name (`hostname`)
  - Set the host name (`mkhostname`).
  - Show a host name (`lshostname`).
- Static Routes (`route`)
  - List all static routes (`lsroute`).
  - Add a static route (`mkroute`).
  - Remove a Static Route (`rmroute`).
- Network Interfaces (`netinterface`)
  - Network Interface Drivers (`ifdriver`)
    - Change/Show Characteristics of a Network Interface Driver (`chif`, `lsif`, `if`).
  - Network Interface Selection (`inet`)
    - List All Network Interfaces (`lsinet`)
    - Add a Network Interface (`mkinet`).
      - Add an Ethernet interface (`mkineten`)
      - Add an IEEE 802.3 interface (`mkinetie3`)
      - Add a Token-Ring interface (`mkinettr`)
      - Add an X.25 interface (`mkinetx25`)
      - Add a Serial Line IP interface (`mkinetsl`).
    - Change or show characteristics of a network interface (`chinet`, `shinet`).

      - Start using syslogd both now and at next system restart (stsyslogd_both).
    - Change/Show characteristics of the syslogd subsystem (chsyslogd, lssyslogd).
    - Stop using the syslogd subsystem (spsyslogd).
      - Stop using the syslogd subsystem now (spsyslogd_now).
      - Stop using the syslogd subsystem at next system restart (spsyslogd_boot).
      - Stop using syslogd both now and at next system restart (spsyslogd_both).
  - Protocols information (protocols).
- Server Network Services (servernet, ruser, startsrc, stopsrc)
  - Remote Access (rmtaccess).
    - Host access (hosts.equiv, hostsequiv).
    - List all remote hosts (lshostsequiv).
    - Add a remote host (mkhostsequiv).
    - Remove a remote host (rmhostsequiv).
  - Restrict File Transfer Program Users (ftpusers)
    - Show all restricted users (lsftpusers).
    - Add a restricted user (mkftpusers).
    - Remove a restricted user (rmftpusers).
- Other Available Services (otherserv)
- Super Daemon (inetd)
  - inetd subsystem (inetdsubsys)
    - Start using the inetd subsystem (mkinetd).
      - Start using the inetd subsystem now (mkinetd_now).
      - Start using the inetd subsystem at next system restart (mkinetd_boot).
      - Start using inetd both now and at next system restart (mkinetd_both).
    - Change/Show characteristics of the inetd subsystem (chinetd, lsinetd).
    - Stop using the inetd subsystem (rminetd).
      - Stop using the inetd subsystem now (rminetd_now).
      - Stop using the inetd subsystem at next system restart (rminetd_boot).
      - Stop using inetd both now and at next system restart (rminetd_both).
  - inetd Subservers (inetd.conf, inetdconf)
    - List all inetd subservers (lsinetdconf).

        - Stop using gated both now and at next system restart (`spgated_both`).

- named Subsystem (`named`).
    - Start using the named subsystem (`stnamed`).
        - Start using the named subsystem now (`stnamed_now`).
        - Start using the named subsystem at next system restart (`stnamed_boot`).
        - Start using named both now and at next system restart (`stnamed_both`).
    - Change/Show characteristics of the named subsystem (`chnamed, lsnamed`).
    - Stop using the named subsystem (`spnamed`).
        - Stop using the named subsystem now (`spnamed_now`).
        - Stop using the named subsystem at next system restart (`spnamed_boot`).
        - Stop using named both now and at next system restart (`spnamed_both`).
- timed Subsystem (`timed`).
    - Start using the timed subsystem (`sttimed`).
        - Start using the timed subsystem now (`sttimed_now`).
        - Start using the timed subsystem at next system restart (`sttimed_boot`).
        - Start using timed both now and at next system restart (`sttimed_both`).
    - Change/Show characteristics of the timed subsystem (`chtimed, lstimed`).
    - Stop using the timed subsystem (`sptimed`).
        - Stop using the timed subsystem now (`sptimed_now`).
        - Stop using the timed subsystem at next system restart (`sptimed_boot`).
        - Stop using timed both now and at next system restart (`sptimed_both`).
- rwhod Subsystem (`rwhod`).
    - Start using the rwhod subsystem (`strwhod`).
        - Start using the rwhod subsystem now (`strwhod_now`).
        - Start using the rwhod subsystem at next system restart (`strwhod_boot`).
        - Start using rwhod both now and at next system restart (`strwhod_both`).
    - Stop using the rwhod subsystem (`sprwhod`).
        - Stop using the rwhod subsystem now (`sprwhod_now`).
        - Stop using the rwhod subsystem at next system restart (`sprwhod_boot`).
        - Stop using rwhod both now and at next system restart (`sprwhod_both`).
- portmap Subsystem information (`portmap`)
- PTYs (`pty`)
    - List all PTY's (`lsall`)
    - Add a PTY (`mkpty`)
    - Change characteristics of a PTY

Planning Your TCP/IP Network

Because AIX TCP/IP is such a flexible networking tool, there are many ways you can customize TCP/IP to fit the specific needs of your organization. Listed below are the major things you need to consider when planning your network.

- Decide which type of network hardware you want to use (for example, token-ring or ethernet).
- Plan the physical layout of the network.

  Consider which functions each host machine will serve. For example, you will need to decide which machine or machines will serve as gateways before you cable the network.
- Decide whether a *flat* network or a *hierarchical* network organization best fits your needs.

  If your system is fairly small, at a single site, and consists of one physical network, then a flat network will probably suit your needs. If your network is very large or complex with multiple sites or multiple physical networks, a hierarchical network may be a more efficient network organization for you.
- If your network will be connected to other networks, plan how your gateways need to be set up and configured. You will need to:
  - Decide which machine or machines will serve as gateways.
  - Decide whether you need to use static or dynamic routing, or a combination of the two. If you choose dynamic routing, decide which routing daemons each gateway will use in light of the types of communications protocols you will need to support.

  See Understanding Gateways for TCP/IP on page 14–39, Configuring Gateways for TCP/IP on page 14–41, and Understanding Routing for TCP/IP on page 14–37 for more information.
- Decide on an addressing scheme.

  If your network will not be part of a larger internetwork, choose the addressing scheme that best fits your needs. If you want your network to be connected to a larger internetwork such as the DARPA Internet, contact their central authority.

  See Understanding Addresses for TCP/IP on page 14–30 for more information.
- Decide whether your system needs to be divided into subnets. If so, decide how you will assign subnet masks. Subnet masks are optional.

  See Subnet Addresses for TCP/IP on page 14–33 for more information.
- Decide on a naming scheme. Each machine on the network will need its own unique host name.

  See Understanding Naming for TCP/IP on page 14–24 for more information.
- Decide whether your network will need a name server for name resolution or if using the **/etc/hosts** file will be sufficient.

  If you choose to use name servers, consider what type of name servers you will need and how many you will need to serve your network efficiently.

  See Understanding Naming for TCP/IP on page 14–24 and Configuring Name Servers for TCP/IP on page 14–29 for more information.

Checklist for Configuring a TCP/IP Network

Prerequisite Tasks or Conditions

1. Network hardware is set up and cabled.
2. TCP/IP software is installed.

Procedure

1. Configure each host machine on the network.

   See How to Configure a Host on a TCP/IP Network on page 14–97 for more information.

   Note: Each machine on the network needs this basic configuration whether it will be an end-user host, a file server, a gateway, or a name server.

2. If you are setting up a hierarchical DOMAIN network, configure at least one host to function as a name server.

   See Understanding Naming for TCP/IP on page 14–24 and Configuring Name Servers on page 14–29 for more information.

3. If your network is to communicate with any remote systems, configure at least one host to function as a gateway. The gateway can use static routes or a routing daemon to perform internetwork routing.

   Configure gateways using routing daemons appropriate to the types of communication you need and the networks you will be communicating with.

   See Understanding Gateways for TCP/IP on page 14–39, Configuring Gateways for TCP/IP on page 14–41, Understanding Routing for TCP/IP on page 14–37, and Static and Dynamic Routing in TCP/IP on page 14–38 for more information.

4. Configure any remote print servers you will need.

   See Remote Services Overview for more information.

5. If desired, configure a host to use or serve as master time server for the network.

   See the **timed** daemon for more information.

Related Information

Understanding Installation and Configuration for TCP/IP on page 14–80.

How to Install AIX for RISC System/6000 Licensed Program Optional Program Products in *General Concepts and Procedures.*

Further Host Configuration

The following procedure provides alternative methods for defining the required initial values as well as information on more advanced host configuration. To set up a new host on a TCP/IP network, some of the following steps are required and some steps are optional.

The following steps are required if you have not performed the minimal configuration procedure (above):

1. Configure the network interface for the adapter using the following SMIT keyword:

   ```
   smit chinet
   ```

   Most of the network interface configuration is done automatically using default values during the first system startup after the adapter is installed. However, you must set the Internet address of the adapter. Depending on the type of network interface and its requirements, you may need to change interface specific information as well.

   See How to Configure a Network Interface on page 14–100 for more information.

2. Set the host name of the new host using the following SMIT keyword:

   ```
   smit mkhostname
   ```

   Using the **mkhostname** command through SMIT changes the host name for the current session and in the configuration database. Use the full domain name here.

At least one of the following steps must be performed:

- To enable the host to use a name server, set up the **resolv.conf** file to include the name, Internet address, and default domain of the name server host. Use the following SMIT keyword:

  ```
  smit stnamerslv
  ```

  See How to Configure a Host to Use a Name Server on page 14–116 for more information.

OR

- Set up the /**etc**/**hosts** file to contain name-to-address mapping for each host on the local network. You can either edit the /**etc**/**hosts** file manually, or use the following SMIT keyword:

  ```
  smit mkhostent
  ```

  Note: You can use the /**etc**/**hosts** file for name resolution even if you are also using a name server. In this case, a host will query the name server first, then check its local /**etc**/**hosts** file if the name server is unable to resolve a name. For example, you might put entries in the /**etc**/**hosts** file for the names of hosts on a local private network for which the name server is not responsible.

The following steps are optional:

- Edit the **rc.tcpip** file format. Remove the comment symbol (#) from any daemons you want to start with each IPL.

OR

- Configure daemons you want to start with each system startup in the configuration database by entering the following command:

  ```
  smit otherserv
  ```

How To Configure a Network Interface

Prerequisite Tasks or Conditions

1. TCP/IP is installed.
2. If using X.25 communications, X.25 software is installed.

Procedure

To configure a network interface, use the following steps:

1. To get to the Change/Show Characteristics of Network Interface menu, enter the following command:

   ```
   smit chinet
   ```

   The system will display a list of interfaces that have been partially and automatically configured by the operating system. There will be one entry for each communications adapter installed in the host. The list will look something like the following:

   ```
   en0
   tr0
   tr1
   ```

2. Select the network interface you want to configure. (The type of interface is designated by the first two characters of the entry: `en` for ethernet, `tr` for token-ring, `sl` for SLIP.)
3. Specify the Internet Address (required) and the Network Mask (optional), both in dotted decimal notation. The Internet Address is a required field for all network interface types.
   a. If you choose a token-ring type interface, you may choose to confine broadcasting to the local token-ring. This is not a required field.
   b. If you choose a SLIP network interface type in step 3, specify the destination address in dotted decimal notation and the TTY instance. If you are using a modem, you can also specify the baud rate and the dial string. This is a required field.

   Note: If you are using the SLIP network interface, you must first create a TTY device.
4. If you chose an X.25 network interface type in step 3, you also need to add IP/X.25 entries to the ODM database. To do so, use the following steps:
   a. From the SMIT Network Interfaces menu, select `IP/X.25 Host Configuration`. Then select `Add an IP/X25 Entry`.
   b. Choose whether you want the host you are adding to have a PVC or an SVC circuit.
   - If you specify a PVC circuit, also specify the `Host Name` and the `Logical Channel Number`. These are both required fields.
   - If you specify an SVC circuit, also specify the Host Name and Remote DTE address. These are both required fields. Specify any of the X.25 optional facilities you would like.

     Note: Any remote host with which you want to set up X.25 communications must either be set up in the local host's **/etc/hosts** file or have name resolution information set up in the name server for the local host's zone of authority.

   c. Repeat steps a and b for each host you want to set up.

How To Configure the inetd Daemon

Prerequisite Condition

1. TCP/IP is installed.
2. Basic configuration has been performed on the host machine. See How to Configure a Host on page 14–97.

Procedure

The **inetd** daemon reduces system load by invoking other daemons only when they are needed. Configure the **inetd** daemon in one of the following ways:

- Update the configuration database for the **inetd** daemon by modifying the **InetServ** object data class. This can be done through SMIT using the **chinet** keyword.

  Note: The **InetServ** object data class can also be edited directly in the configuration database using the ODM editor. If the object data class is edited directly, the **inetexp** command must be run to export the data into the **/etc/inetd.conf** and **/etc/services** files.

OR

- Edit the **inetd.conf** file format, then run the **inetimp** command to import the data into the **InetServ** object class.

Related Information

The **inetimp** command, **inetexp** command.

The **inetd** daemon.

The **inetd.conf** file, **/etc/services** file.

Understanding the SMIT Interface for TCP/IP on page 14–88.

- Set up the egpmaxacquire stanza to specify a maximum number of EGP peers with which your network's gateway may exchange information. Use EGP options to set attributes for each neighbor specified. If this stanza is omitted, all EGP neighbors are acquired. For example:

```
EGP     yes
EGP Options
autonomoussystem     546
egpneighbor          earl
   nogendefault
egpmaxacquire        5
```

If using RIP:

- Set the RIP stanza to yes.
- Enter quiet into the RIP stanza if you want the gateway only to accept routing information, not broadcast information. Or enter supplier into the RIP stanza if you want the gateway to broadcast routing information as well as accept routing information.
- Enter pointopoint if you want the gateway to broadcast only to source gateways specified in the sourceripgateways stanza. If not, omit this value.
- If you include pointopoint, enter a gateway name or Internet address in dotted decimal, along with a default metric. For example:

```
RIP   yes   supplier   101.25.32.1   2
```

If using HELLO:

- Set the HELLO stanza to yes.
- Enter quiet into the HELLO stanza if you want the gateway only to accept routing information, not broadcast. Or enter supplier into the HELLO stanza if you want the gateway to broadcast routing information as well as accept routing information.
- Enter pointopoint if you want the gateway to broadcast only to another gateway. If not, omit this value.
- If you include pointopoint, enter a gateway name or Internet address in dotted decimal, along with a default metric. For example:

```
HELLO   yes   quiet   101.25.32.1   3
```

If using SNMP:

- Set the SNMP stanza to yes.
- Make sure that **mib** is set up in the **/etc/services** file as a UDP service.

- If desired, specify by protocol the networks and hosts that are announced or not announced. Use the following stanzas:

  announce *Network InterfaceAddress* [*Address* ...] *ProtocolType* [*EGPMetric*]

  announcehost *Host InterfaceAddress ProtocolType* [*EGPMetric*]

  noannounce *Network InterfaceAddress* [*Address* ...] *ProtocolType* [*EGPMetric*]

  noannouncehost *Host InterfaceAddress ProtocolType* [*EGPMetric*]

- If desired, specify EGP networks to restrict. This stanza cannot be used with the **announce/noannounce** stanza. Use the following stanza:

  egpnetsreachable *Network* [*Network Network* ...]

- If desired, specify known invalid networks to ignore. Use the following stanza:

  martiannets *Network* [*Network Network* ...]

- If desired, use the following stanza for network validation:

  validAS *Network* **AS** *System* **metric** *Number*

d. If using EGP, specify a default metric and a default gateway. Use the following stanzas:

   - **defaultegpmetric** *Number*
   - **defaultgateway** *Gateway* [*Metric*] *Protocol* [**active** | **passive**]

e. For each protocol you have set to `yes`, use the `net` and `host` stanzas to set up any static routes. The net stanza installs a route to a network address, and the host stanza installs a route to a host address. Static routes are not updated by the **gated** daemon; manual input is required to update a static route.

Related Information

The **gated** daemon, **routed** daemon.

The **/etc/networks** file, **gated.conf** file.

Configuring Gateways for TCP/IP on page 14–41.

Understanding Gateways for TCP/IP on page 14–39, Understanding Protocols for TCP/IP on page 14–54, Understanding Routing for TCP/IP on page 14–37, Understanding SRC Control for TCP/IP Daemons on page 14–21.

How to Configure the routed Daemon on page 14–107.

Planning for DOMAIN Name Resolution

If you are part of a larger internetwork, you will need to coordinate setting up your domain and name servers with their central authority.

Some hints in planning your own domain name resolution system:

1. Plan ahead.

   Changing a name is *much* more difficult than setting up the initial one. Get consensus from your organization on network, gateway, name server, and host names before you set up your files.

2. Set up redundant name servers.

   If you cannot set up redundant name servers, be sure to set up secondary and cache name servers so you have some type of backup.

3. In selecting the name servers, keep the following in mind:

   a. Choose machines for name servers that are physically closest to exterior systems.

   b. The name servers should be as independent as possible. Try for different power supplies and independent cabling.

   c. Find another network to back up your name resolution service, and do the same for them.

4. Test the servers.

   a. Test both regular and reverse name resolution.

   b. Test zone transfer from primary to secondary name servers.

   c. Test each name server after a system crash and reboot.

5. Send name resolution requests to forwarder servers before they go to exterior name servers. This will allow your name servers to share caches and improve performance by reducing the load on your primary name servers.

Related Information

The **named** daemon .

The DOMAIN Local Data file, DOMAIN Cache file, DOMAIN Data file, DOMAIN Reverse Data file, **named.boot** file.

Configuring Name Servers for TCP/IP on page 14–29.

How to Configure a Primary Name Server on page 14–109, How to Configure a Secondary Name Server on page 14–112, How to Configure a Cache-Only Name Server on page 14–114, How to Configure a Host to Use a Name Server on page 14–116.

Understanding Naming for TCP/IP on page 14–24.

3. Edit the local data file. See the DOMAIN Local File for a detailed example of a local data file.

   a. Specify the Start of Authority of the zone and the default time-to-live information. For example:

      ```
      @   IN   SOA    venus.abc.aus.ibm.com. gail.zeus.abc.aus.ibm.com.
      (
                                    1.1      ;serial
                                    3600     ;refresh
                                    600      ;retry
                                    3600000 ;expire
                                    86400)   ;minimum
      ```

   b. Specify the Name Server (NS) record. For example:

      ```
              IN     NS      venus.abc.aus.ibm.com.
      ```

   c. Specify the Pointer (PTR) record.

      ```
      1       IN     PTR     localhost.
      ```

      Note: All lines in this file must be in Standard Resource Record Format.

4. Edit the hosts data file. See the DOMAIN Data file for an example of a hosts data file.

   a. Specify the Start of Authority of the zone and the default time to live information for the zone. This record designates the start of a zone. There should only be one Start of Authority Record per zone. For example:

      ```
      @    IN    SOA    venus    bob.robert.abc.aus.ibm.com.
      (
                                    1.1      ;serial
                                    3600     ;refresh
                                    600      ;retry
                                    3600000 ;expire
                                    86400)   ;minimum
      ```

   b. Include name-to-address resolution information on all hosts in the name server's zone of authority. For example:

      ```
      venus                     IN    A        192.9.201.1
      venus                     IN    A        128.114.100.1
      ```

   c. Include Name Server Records for all primary name servers in the zone. For example:

      ```
                                 IN    NS       venus.abc.ibm.com
                                 IN    NS       kronos.xyz.ibm.com
      ```

   d. Include other types of entries, such as Canonical Name Records and Mail Exchanger Records as needed (optional).

      Note: All lines in this file must be in Standard Resource Record Format.

Related Information

The **/etc/hosts** file, **named.boot** file, DOMAIN Cache file, DOMAIN Local file, DOMAIN Data file, DOMAIN Reverse Data file, **resolv.conf** file, **rc.tcpip** file.

The **named** daemon.

Configuring Name Servers for TCP/IP on page 14–29.

How to Configure a Secondary Name Server on page 14–112, How to Configure a Cache-Only Name Server on page 14–114, How to Configure a Host to Use a Name Server on page 14–116.

The Standard Resource Record Format.

How To Configure a Secondary Name Server

Prerequisite Tasks or Conditions

1. TCP/IP is installed.
2. Basic configuration has been performed on the host machine. See How to Configure a Host on page 14–97.

Procedure

Configure a secondary name server according to the following steps:

1. Edit the **named.boot** file. See the **named.boot** file for a detailed example of a boot file.

   This file is read each time the **named** daemon starts. It tells the server what type of server it is, the zone for which it is responsible, and where to get its initial information.

   a. Specify the directory the **named** data files can be located (optional). Use this line if you want the **named** data files to use paths relative to this directory. For example:

   ```
   directory /usr/local/domain
   ```

   b. Specify the default domain for the name server. For example:

   ```
   domain        abc.aus.ibm.com
   ```

   c. Specify the name of the cache file for the name server. For example:

   ```
   cache         .                    /etc/named.ca
   ```

   d. Include a secondary line to define the domain for which the name server will be responsible and the Internet addresses from which the host is to get its initial data. Include a line for each domain for which the host is a back-up name server. Specify a file name to which the name server can back up its data. For example:

   ```
   secondary   abc.aus.ibm.com   192.9.201.1   192.9.201.2\
   /etc/named.abc.bak
   secondary   xyz.aus.ibm.com   192.9.201.1   192.9.201.2\
   /etc/named.xyz.bak
   ```

   e. Include a secondary line to define the reverse name resolution information for the name server. For example:

   ```
   secondary   201.9.192.in–addr.arpa 192.9.201.1 192.9.201.2\
   named.rev.bak
   secondary   100.114.128.in–addr.arpa 192.9.201.1 192.9.201.2\
   named.rev.bak
   ```

6. If you chose not to initialize the **named** daemon through SMIT, start the daemon for this session by issuing the following command:

```
startsrc –s named
```

Related Information

The **named** daemon.

The **/etc/hosts** file, **named.boot** file, DOMAIN Cache file, DOMAIN Local file, DOMAIN Data file, DOMAIN Reverse Data file, **resolv.conf** file, **rc.tcpip** file.

Configuring Name Servers for TCP/IP on page 14–29.

How to Configure a Primary Name Server on page 14–109, How to Configure a Cache-Only Name Server on page 14–114, How to Configure a Host to Use a Name Server on page 14–116.

Understanding Naming for TCP/IP on page 14–24.

The Standard Resource Record Format.

How To Configure a Cache-Only Name Server

Prerequisite Tasks or Conditions

1. TCP/IP is installed.
2. Basic configuration has been performed on the host machine. See How to Configure a Host on page 14–97.

Procedure

Configure a cache name server according to the following steps:

1. Edit the **named.boot** file. See the **named.boot** file for a detailed example of a boot file.
   - Specify a name server type of *primary* with a source of **/etc/named.local** as well as the domain for which the name server will be responsible. For example:

     ```
     primary        0.0.127.in–addr.arpa    /etc/named.local
     ```

   - Define the name of the cache file. For example:

     ```
     cache          .                       /etc/named.ca
     ```

2. Edit the cache file. See the DOMAIN Cache file for a detailed example of a cache file.

   This file contains the addresses of the servers that are authoritative name servers for the root domain of the network. For example:

   ```
   ; root name servers.
            1          IN    NS    relay.ibm.com.
   relay.ibm.com.   3600000    IN    A     129.114.1.2
   ```

   Note: All lines in this file must be in Standard Resource Record Format.

Related Information

The **named** daemon.

The /etc/**hosts** file, **named.boot** file, DOMAIN Cache file, DOMAIN Local file, DOMAIN Data file, DOMAIN Reverse Data file, **resolv.conf** file, **rc.tcpip** file.

Configuring Name Servers for TCP/IP on page 14–29.

How to Configure a Primary Name Server on page 14–109, How to Configure a Secondary Name Server on page 14–112, How to Configure a Host to Use a Name Server on page 14–116.

Understanding Naming for TCP/IP on page 14–24.

The Standard Resource Record Format.

How to Configure a Host to Use a Name Server

Prerequisite Tasks or Conditions

1. AIX TCP/IP is installed.
2. Basic configuration has been performed on the host machine. See How to Configure a Host on page 14–97.
3. The name server for host's domain is configured.

Procedure

1. Create the /**etc/resolv.conf** file with the following entries:
   - The full name of the domain for which the name server is responsible. For example:

     ```
     domain abc.aus.ibm.com
     ```

   - Up to 16 name server entries (optional). These entries specify the Internet address of the name server to be queried. The system will query the name servers in the order listed. For example:

     ```
     nameserver 192.9.201.1
     nameserver 192.9.201.2
     ```

2. Test the communication between the host and the name server using the following command:

   ```
   ping hostname
   ```

   Use the name of a host that should be resolved by the name server to see if the process is working.

Related Information

The **resolv.conf** file.

Configuring Name Servers for TCP/IP on page 14–29.

How to Configure a Host on a TCP/IP Network on page 14–97.

Name Resolution Problems in TCP/IP

Resolver routines on hosts running TCP/IP attempt to resolve names using the following sources in the order listed:

1. The DOMAIN (**named**) name server
2. The local **/etc/hosts** file.

If you are using IBM NFS network information service (NIS) for name resolution, resolver routines will attempt to resolve names using the following sources in the order listed:

1. The DOMAIN (**named**) name server
2. Network information service.

If you are using network information service for name resolution, the local **/etc/hosts** file will not be consulted.

On a Client Host

If you cannot get a host name resolved:

If you are using flat name resolution (using the **/etc/hosts** file):

- Verify that the host name and correct IP address information is in the **/etc/hosts** file.

If you are using a name server:

- Verify that you have a **resolv.conf** file specifying the domain name and Internet address of a name server.
- Verify that the local name server is up by issuing the **ping** command with the IP address of the name server (found in the local **resolv.conf** file).
- If the local name server is up, verify that the **named** daemon on your local name server is active by issuing the **ping** command with the IP address of the name server (found in the local **resolv.conf** file).
- If you are running the **syslogd**, there could be error messages logged. The output for these messages is defined in the **/etc/syslog.conf** file.

If these steps do not identify the problem, start looking at the name server host.

On a Name Server Host

If you cannot get a host name resolved:

- Verify that the **named** daemon is active by issuing the following command:

  ```
  lssrc –s named
  ```

- The address of the target host does not exist or is incorrect in the name server's database. Send a SIGINT signal to the **named** daemon to dump the database and cache to the **/usr/tmp/named_dump.db** file. Verify that the address you are trying to resolve is there and is correct.

  Add or correct name to address resolution information in the **named** Hosts Data file for the primary name server of the domain. Then issue the following SRC command to re-read the datafiles:

  ```
  refresh –s named
  ```

Routing Problems in TCP/IP

If you cannot reach a destination host:

If you receive a "Network Unreachable" error message:

- Make sure that a route to the gateway host has been defined and is correct. Check this by using the **netstat –r** command to list kernel routing tables.

If you receive a "No route to host" error message:

- Verify that the local network interface is up by issuing the **netstat –i** command. Use the **ping** command to try and reach another host on your network.

If you receive a "Connection timed out" error message:

- Verify that the local gateway is up using the **ping** command with the name or Internet address of the gateway.
- Make sure that a route to the gateway host has been defined and is correct. Check this by using the **netstat –r** command to list kernel routing tables.
- Make sure the host you want to communicate with has a routing table entry back to your machine.

If you are using static routing:

- Make sure that a route to the target host has been defined. Check this by using the **netstat –r** command to list kernel routing tables.
- Make sure that a route to the gateway host has been defined. Check this by using the **netstat –r** command to list kernel routing tables.

  Note: Make sure the host you want to communicate with has a routing table entry to your machine.

If you are using dynamic routing:

- Verify that the gateway is listed and correct in the kernel routing tables by issuing the **netstat –r** command.

If the gateway host is using the RIP protocol with **routed**:

- Make sure that a static route to the target host is set up in the /etc/**gateways** file.

  Note: You only need to do this if the routing daemon cannot identify the route to a distant host through queries to other gateways.

If the gateway host is using the RIP protocol with **gated**:

- Make sure that a static route to the target host is set up in the **gated.conf** file.

If you are using dynamic routing with the routed daemon:

- If **routed** cannot identify the route through queries (for example, if the target host is not running the RIP protocol) check the /etc/**gateways** file to verify that a route to the target host is defined.
- Make sure that gateways responsible for forwarding packets to the host are up and running the RIP protocol. Otherwise, you'll need to define a static route.

Common Problems with TCP/IP Network Interfaces

Network Interfaces are configured automatically during the first IPL after the adapter card is installed. However, there are certain values that must be set in order for TCP/IP to start. These include the hostname and Internet address and can be set using the **mktcpip** command from the command line. (Use the **mktcpip** command to set these values permanently in the configuration database. Use the **ifconfig** and **hostname** commands to set them in a running system for the current session.) If you have already checked these to verify accuracy and you are still having trouble sending and receiving information, check the following:

- Verify that the Network Interface was loaded prior to use by issuing the **netstat –i** command. If it is not listed, access the SMIT dialog to initialize the network interface by issuing the following command:

  ```
  smit mkinet
  ```

  You can also initialize the network interface in the running system for the current session using the **ifconfig** command. Making changes using the **ifconfig** command will not update the configuration database.

- Make sure the interface address was set correctly in the interface configuration by issuing the **netstat –i** command. If it is incorrect, access the SMIT dialog to modify it by running the following command:

  ```
  smit chinet
  ```

  You can set the interface address in the running system for the current session using the **ifconfig** command.

- Check the error log by running the **errpt** command to ensure that there are no adapter problems.

- Verify that the adapter card is good by running diagnostics.

If these steps do not identify the problem, refer to Problems with a SLIP Network Interface, Problems with an X.25 Interface, Problems with an Ethernet Network Interface, or Problems with a Token-Ring Network Interface.

Problems with a SLIP Network Interface

If the network interface has been initialized, the addresses correctly specified, and you have verified that the adapter card is good:

- Verify that the **slattach** process is running and using the correct TTY port by issuing the **ps –ef** command. If it is not, run the **slattach** command to change the SLIP line discipline and deny access to the port by other applications.

- Verify that the point to point addresses were specified correctly in the interface configuration by running the **netstat –r** command. If the addresses are incorrect, access the SMIT dialog to correct them using the following command:

  ```
  smit chinet
  ```

  You can set the point to point addresses in the running system for the current session using the **ifconfig** command.

Problems with a Token–Ring Network Interface

If you cannot communicate with some of the machines on your network and the network interface has been initialized, the addresses correctly specified, and you have verified that the adapter card is good:

- Check to see if the hosts you cannot communicate with are on a different ring. If they are, you may have the *allcast* attribute set incorrectly. The *Confine Broadcast to Local Token-Ring* field should be set to *no* in the token-ring network interface SMIT dialog. Use the following command to access the SMIT Network Interfaces menu:

  ```
  smit chinet
  ```

Related Information

The **errpt** command, **mktcpip** command, **netstat** command, **ps** command, **slattach** command, **x25xlate** command.
The SMIT Overview in *General Programming Concepts.*

Problems with Packet Delivery in TCP/IP

If you are having trouble with packet loss or are experiencing delays in packet delivery, try the following:

- Use the **trpt** command to trace packets at the socket level.
- Use the **iptrace** command to trace all protocol layers.

Refer to the **trpt** and **iptrace** commands for more information on values returned by the different types of tracing.

Problems with SRC Support in TCP/IP

If changes to the /etc/inetd.conf file do not take effect:

- Run the **inetimp** command to import the changes into the InetServ object class. Then update **inetd** by issuing the **refresh –s inetd** command (the preferred method), or the command **kill –1 | –2** *<inetd pid>*.

If the startsrc –s [subsystem name] returns the following error message:

```
0513–00  The System Resource Controller is not active.
```

- The System Resource Controller subsystem has not been activated. Issue the **srcmstr &** command to start SRC, then reissue the **startsrc** command.
- You may also want to try starting the daemon from the command line without SRC support.

If the refresh –s [subsystem name] or lssrc –ls [subsystem name] returns the following error message:

```
[subsystem name] does not support this option.
```

- The subsystem does not support the SRC option issued. Check the subsystem documentation to verify options the subsystem supports.

If the following message is displayed:

```
SRC was not found, continuing without SRC support.
```

- A daemon was invoked directly from the command line instead of using the **startsrc** command. This is not a problem. However, SRC commands such as **stopsrc** and **refresh** will not manipulate a subsystem that is invoked directly.

Related Information

The **inetimp** command, **lssrc** command, **refresh** command, **startsrc** command.
The **inetd** daemon.
The **inetd.conf** file.
Understanding SRC Control of TCP/IP Daemons.

TCP/IP Configuration Problems

Network Interfaces are automatically configured during the first IPL after the adapter card is installed. However, you still need to set some initial values for TCP/IP including the host name, the Internet address, and the subnet mask. Use the **mktcpip** command to set these initial values. The **mktcpip** command will also allow you to specify a name server to provide name resolution service. Other network interface attributes must be set from the SMIT Network Interface menu. Use the following command to access this menu:

```
smit chinet
```

Note: The **mktcpip** command will only allow you to configure one network interface. Configure additional network interfaces using the following command:

```
smit chinet
```

You may also want to set up any static routes the host will need for sending transmitting information, such as a route to the local gateway. To set these up permanently in the configuration database, use the following command:

```
smit mkroute
```

If you are having other problems with your configuration, see the Checklist for Configuring Your TCP/IP Network for more information.

Related Information

The **mktcpip** command, **rc.net** file.
The SMIT Overview in *General Programming Concepts*.

Real-Time Control Microcode

See Installing the Real-Time Control Microcode (RCM) on page 15–4.

Application Programming Interface

See the X.25 Overview for Programming in the *Communications Programming Concepts and Procedures* book.

X.25 Commands

See the X.25 Commands Overview on page 15–36.

X.25 Device Driver

The X.25 device driver handles communication between the X.25 adapter code and applications such as:

- Applications using the X.25 API subroutines
- SNA applications using QLLC
- Applications using TCP/IP.

Qualified Logical Link Control (QLLC)

QLLC is a data link control that enables SNA-to-SNA connections over the X.25 network. For more information, see Understanding Qualified Logical Link Control (DLCQLLC).

Managing X.25

Managing X.25 communications involves several tasks. You use the **smit** command for some of these tasks; other tasks can be performed using special the X.25 commands, **xcomms**, **xroute**, **xtalk**, **xmanage**, and **xmonitor**.

Initial System Management

Before you can use X.25, you need:

An X.25 line and network terminating unit (NTU)
An X.25 adapter and a cable.

Then you need to:

Install the X.25 software
Configure X.25
Verify that the set-up has succeeded.

Regular System Management

To enable users to receive incoming calls and make outgoing calls:

Make changes to the routing list, using the **xroute** command
Make changes to the system address list, using the **xtalk** command.

You can use the **xmanage** command to:

Display status of an X.25 port (connected/disconnected)
Connect or disconnect an X.25 port while the system is running.

To get better performance, you may have to tune X.25 communications:

Get statistics for an X.25 port, using the **xmanage** command
Monitor activity on an X.25 port, using the **xmonitor** command
If necessary, change the attributes of the X.25 adapter, using the **smit** command.

You may also have to diagnose problems from time to time.

X.25 Installation and Configuration Overview

This is what you have to do *before* you can use X.25 communications:

Get an X.25 line and network terminating unit (NTU)
Get an X.25 adapter and a cable
Install the real-time control microcode
Install the X.25 software
Configure X.25
Add entries to the X.25 routing list (optional)
Verify that the set-up has succeeded.

Getting an X.25 Line and Network Terminating Unit (NTU)

Before you can use X.25 communications, you need to subscribe to a line to a public or private X.25 network.

Network lines are available through subscription from a Post, Telegraph, and Telephone (PTT) authority or network provider; they may take some time to obtain, so you should order one well in advance. When you apply, you choose various options, such as the number of logical channels required. Some of these options are also specified when you configure the network attributes.

Your network provider gives you a network user address when you subscribe to a public network.

Getting an X.25 Adapter and a Cable

You need an X.25 adapter, installed in the RISC System/6000. The cable (X.21 *bis* (V.24 or V.35) or X.21) is ordered separately.

Installing the Real-Time Control Microcode (RCM)

Each X.25 adapter comes with a diskette containing the real-time control microcode (RCM). If the software was preloaded at the factory, the X.25 code will already be on the hard disk. If you ordered an X.25 adapter at a later date, the RCM will need to be installed. If possible you should install the RCM before installing the adapter. (If you install the adapter first, when you start up the system you will see error messages from the X.25 configuration; ignore these messages until you have installed the RCM.)

To install the RCM, insert the RCM diskette and enter:

```
adfutil —d/dev/fd0 —a x25s —m —q
```

If you have problems with configuring the X.25 adapter, check the error log, which might indicate that the RCM has not been transferred successfully to the system from its diskette.

Configuring the X.25 Adapter

The installation process added a definition for the X.25 adapter. Without some information from you, however, the adapter cannot be used for communications. You must add the network user address (NUA) and the network identifier; this sets default values for several of the other X.25 adapter attributes. These default values are suitable for the network indicated by the country code part of the NUA and the network identifier.

When completed, the initial configuration gives a partially working, but not fully configured system. You may have to change some of the network attributes to values supplied by your network provider (particularly if the network cannot be identified from the NUA).

You may also have to change some of the packet attributes and define permanent virtual circuits (PVCs), if you intend to use them.

For more information, see How to Complete the Initial Configuration of the X.25 Adapter on page 15–55.

Adding Entries to the X.25 Routing List

The routing list is supplied with default entries so that you can receive incoming calls right away. There are entries suitable for using X.25 with SNA and TCP/IP (**IBMQLLC** and **IBMTCPIP**, respectively). Nevertheless, you might have your own requirements that can be met only if you add new entries to the routing list.

Verifying that the Set-Up has Succeeded

As soon as you have set up X.25 communications, you need to run your application to verify that you can communicate over X.25. You could use the **xmanage** command to check your connection to the network. Then you could use the **xtalk** command to make a call to another user, exchange messages, and send a file.

Related Information

How to Configure the X.25 Adapter on page 15–55.
How to Manage the X.25 Routing List on page 15–59.
How to Display Status Information for an X.25 Port on page 15–74.
xtalk Overview on page 15–24.
X.25 Problem Diagnosis Overview on page 15–25.
Customizing SNA Services on page 13–45.
How to Configure a Network Interface on page 14–100.
X.25 Communications Overview for System Management on page 15–1.

List of X.25 Network Configuration Attributes

You *must* set the network user address (NUA), before you can use X.25 communications. You may have to set values for other attributes listed here to values supplied by your network provider.

Network identifier

Indicates the network you intend to use. The value can be Datex-P, Datapac, Telenet, DDN, other public, or other private. The default value is other public.

Local network user address (NUA)

Indicates the network user address (NUA) of the local data terminal equipment (DTE). Do *not* include any subaddress (suffix). The value must be 1 through 15 decimal digits.

Lowest logical channel number for an incoming SVC

Indicates the lowest-numbered logical channel that can be used for an incoming switched virtual circuit (SVC). The number depends on your network subscription and must be in the range 0 through 4095. The default value is 0.

Number of logical channels for incoming SVCs

Indicates the maximum number of logical channels that can be used for incoming switched virtual circuits (SVCs). The number depends on your network subscription and must be in the range 0 through 64. The total number of virtual circuits for this adapter must not exceed 64. The default value is 0.

Lowest logical channel number for a two-way SVC

Indicates the lowest-numbered logical channel that can be used for a two-way switched virtual circuit (SVC). The number depends on your network subscription and must be in the range 0 through 4095. The default value is 1.

Number of logical channels for two-way SVCs

Indicates the maximum number of logical channels that can be used for two-way switched virtual circuits (SVCs). The number depends on your network subscription and must be in the range 0 through 64. The default value is 20. The total number of virtual circuits for this adapter must not exceed 64.

Lowest logical channel number for an outgoing SVC

Indicates the lowest-numbered logical channel that can be used for an outgoing switched virtual circuit (SVC). The number depends on your network subscription and must be in the range 0 through 4095. The default value is 251.

Number of logical channels for outgoing SVCs

Indicates the maximum number of logical channels that can be used for outgoing switched virtual circuits (SVCs). The number depends on your network subscription and must be in the range 0 through 64. The total number of virtual circuits for this adapter must not exceed 64. The default value is 0.

List of X.25 Packet Configuration Attributes

Check that the default values for the following attributes suit the needs of your system; if not, you must change them.

CCITT support
Indicates which version of X.25 should be supported. The value must be 1980 or 1984. The default is set up by completing the initial configuration.

Packet modulo
Indicates the modulo to be used for packet transmission. The value must be 8 or 128. The default value is 8.

Type of line Indicates whether the adapter is to be configured as a data circuit-terminating equipment (DCE) or a data terminal equipment (DTE). The value must be DCE or DTE. The default value is DTE.

Information about the other packet attributes is grouped as follows:

Default Attributes for Switched Virtual Circuits
Maximum Negotiable Attributes for Switched Virtual Circuits
Optional Facilities Control
ISO8208-Defined Timers
Packet-Level Features.

Default Attributes for Switched Virtual Circuits

Default receive packet size
Indicates the default receive packet size, in bytes, that will be received. This can be altered for an individual call, using the optional facilities in the call-request packet. The size must be one of the following values: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096. The default value is 128.

Default transmit packet size
Indicates the default transmit packet size, in bytes, that will be sent. This can be altered for an individual call, using the optional facilities in the call-request packet. The size must be one of the following values: 16, 32, 64, 128, 256, 512, 1024, 2048, 4096. The default value is 128.

Default receive packet window
Indicates the default receive packet window (the number of packets allowed to be received without confirmation). The number must be in the range 1 through 127. The default value is 2.

Default transmit packet window
Indicates the default transmit packet window (the number of packets allowed to be sent without confirmation). The number must be in the range 1 through 127. The default value is 2.

Default receive throughput class
Indicates the default receive throughput class for non-negotiating calls. The class must be one of the following values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000. The default value is 9600.

Default transmit throughput class
Indicates the default transmit throughput class for non-negotiating calls. The class must be one of the following values: 75, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 48000 . The default value is 9600.

Closed user group (CUG) extended format

This is an optional facility available on some X.25 networks that conform to 1980 or 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

CUG with OA selection basic format

This is an optional facility available on some X.25 networks that conform to 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

CUG with OA selection extended format

This is an optional facility available on some X.25 networks that conform to 1980 or 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Bilateral closed user group (BCUG)

This is an optional facility available on some X.25 networks that conform to 1980 or 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Reverse charging and fast select

This is an optional facility available on some X.25 networks that conform to 1980 or 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Network user identification (NUI)

This is an optional facility available on some X.25 networks that conform to 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Charging requesting service

This is an optional facility available on some X.25 networks that conform to 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Receiving information about monetary unit

This is an optional facility available on some X.25 networks that conform to 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Receiving information about segment count

This is an optional facility available on some X.25 networks that conform to 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

Receiving information about call duration

This is an optional facility available on some X.25 networks that conform to 1984 standards. If your network supports this facility *and* you have subscribed to the facility, select enable; otherwise select disable. The default value is enable.

ISO8208-Defined Timers

T21 timer Indicates the time (in seconds) within which a call-connected, clear-indication, or incoming-call packet should be received following a call-request packet being transmitted. The value must be in the range 1 through 255. The default value is 200.

T22 timer Indicates the time (in seconds) within which a reset-confirmation packet should be received following a reset-request packet being transmitted. The value must be in the range 1 through 255. The default value is 180.

T23 timer Indicates the time (in seconds) within which a clear-confirmation packet should be received following a clear-request packet being transmitted. The value must be in the range 1 through 255. The default value is 180.

T24 timer Indicates the time (in seconds) within which an RR packet should be sent following a packet carrying an acknowledgement being transmitted. The value must be either 0 (meaning T24 is disabled) or in the range 1 through 255. The default value is 180.

T25 timer Indicates the time (in seconds) within which, if no acknowledgment has been received, a packet is retransmitted. This is the packet equivalent of the T1 timer. The value must be either 0 (meaning T25 is disabled) or in the range 1 through 255. The default value is 180.

T26 timer Indicates the time (in seconds) within which an interrupt-confirmation packet should be received following an interrupt packet. After this time, the virtual circuit is reset. The value must be either 0 (meaning T26 is disabled) or in the range 1 through 255. The default value is 180.

Packet-Level Features

Throughput-class negotiation

Indicates whether the throughput class is negotiated or validated. Validate means that, instead of throughput class being negotiated, the incoming throughput class value will be checked for acceptability; if unacceptable, the call will be cleared. The value must be validate or negotiate. The default value is negotiate.

Packet-size negotiation

Indicates whether the packet size is negotiated or validated. Validate means that, instead of packet size being negotiated, the incoming packet size value will be checked for acceptability; if unacceptable, the call will be cleared. The value must be validate or negotiate. The default value is negotiate.

Incoming calls

Indicates whether incoming calls are allowed. If incoming calls are not allowed, they are cleared immediately. The value must be allow or forbid. The default value is allow.

Outgoing calls

Indicates whether outgoing calls are allowed. If outgoing calls are not allowed, they are cleared immediately, returning a clear-indication packet to the user. The value must be allow or forbid. The default value is allow.

Fast select Indicates whether fast-select calls are allowed. The value must be enable or disable. The default value is enable.

D-bit Indicates whether the use of the D-bit in the call-request packet is allowed. The value must be allow or forbid. The default value is allow.

List of X.25 Frame Configuration Attributes

The defaults for these attributes have been set up so that they will satisfy the needs of most networks. You should change their values only if you are sure they need changing and you understand X.25 packet-level and frame-level protocols.

Frame window size

Indicates the window size to use at the frame level. This is the allowable number of frames to be sent or received before waiting for an acknowledgement. If the frame modulo is set to 8, the value must be in the range 1 through 7. If the frame modulo is set to 128, the value must be in the range 1 through 127. The default value is 7.

T1 timer

Indicates the time (in units of 50 milliseconds) after which, if it has not been acknowledged, a frame is retransmitted. This is the CCITT T1 value. The value must be in the range 1 through 255. The default value is 60 (3 seconds).

T4 timer

Indicates the time (in seconds) after which, if there has been no activity on a link, an RR frame is sent. Frame-level recovery is started if no answer is received within T1. The value must be either 0 (meaning T4 is disabled) or in the range 4 through 255. The default value is 180 (3 minutes).

N2 counter

Indicates the maximum number of times a frame can be transmitted under error conditions. This is the CCITT N2 value. The value must be in the range 1 through 255. The default value is 20.

Connection mode

Indicates the connection mode. In passive connection mode, X.25 waits for an SABM from the network to determine whether the network is connected. In active connection mode, X.25 sends SABMs to the network, waiting for the network to send a UA to acknowledge that it is connected. The value must be active or passive. The default is set up by the installation process.

Physical level startup counter

When connecting to the network terminating unit (NTU), DTR is raised and the lines DCD and DSR are tested. These lines becoming active indicates that the network terminating unit is ready. After raising DTR, these lines are tested at intervals and this value gives the number of tests after which the attempted connection is considered to have failed. The value must be in the range 1 through 255. The default value is 11.

Physical level poll timer

Indicates the time (in units of 50 milliseconds) that elapses between each poll of the physical level. The value must be in the range 10 through 255. (10 represents 0.5 second.) The default value is 10.

List of X.25 General Configuration Attributes

The defaults for these attributes have been set up so that they will satisfy the needs of most networks. You should change their values only if you are sure they need changing and you understand X.25 packet-level and frame-level protocols.

Receive data transfer offset
Indicates the receive data transfer offset to be used by SNA services. Do not set this value to anything other than 0 unless you are using SNA. The value must be in the range 0 through 1024. The default value is 0.

Line monitor buffer maximum bytes
Indicates the maximum number of bytes of data that will be traced on the line. The value must be in the range 1 through 65535. The default value is 256.

Line monitor flow control
Indicates whether line monitoring system (LMS) should control the flow of traffic through the X.25 port when there is a shortage of trace buffers. The value must be on or off. The default value is off.

Calling address in call-request packet
Indicates whether the X.25 adapter should pass the calling address when sending a call-request packet. The value must be either allow or forbid. The default value is allow.

Auto-call unit disconnection timeout
Indicates the number of seconds that will elapse between the last call being cleared and the line being disconnected. This only applies if an auto-call unit is attached. The value must be in the range 0 through 1800. (0 means that the line will stay connected, even when there are no calls in progress.) The default value is 0.

Related Information

How to Change X.25 Adapter Attributes on page 15–57.
List of X.25 Network Configuration Attributes on page 15–8.
List of X.25 PVC Configuration Attributes on page 15–19.
List of X.25 Frame Configuration Attributes on page 15–16.
List of X.25 General Configuration Attributes on page 15–18.
X.25 Configuration Attributes Overview on page 15–7.
X.25 Packet Switching Overview on page 15–30.
X.25 Communications: Bibliography on page 15–3.

X.25 Routing Overview

X.25 uses the network user address to route both incoming and outgoing calls to the correct system.

Outgoing Calls

The **xtalk** command uses address lists to route outgoing calls: one list for the system, and a local list for each user. The system address list is available to all users. The individual users can override the names used in the system address list and add other names and addresses to their local lists.

Application programs that make outgoing calls do *not* use these address lists. TCP/IP and SNA have their own methods for making outgoing calls.

Incoming Calls

All incoming calls are routed using the X.25 routing list. Once a call has reached a system, it is routed to an application. An application listens for the incoming calls it is designed to handle by specifying the name of an entry in the routing list. The name of the entry is an arbitrary name known to the application program. The entry specifies the criteria that must be satisfied for the application to handle this call. For example, the criteria can include a specific subaddress, so that an application can listen for calls for a single user.

The routing list entry also includes a user name (the login name). A person whose user name does not match this user name is not allowed to listen for calls for this routing list entry. For example, here are some of the details from one routing list entry:

```
Name:       X25GUEST
User name:  guest
```

A person whose user name is `guest` can listen for calls for this routing list entry, but no other user can. `X25GUEST` is the identifier used by the application program when it starts listening.

Forwarding or Rejecting Incoming Calls

What happens to an incoming call that matches a routing list entry for which no application is listening? This depends on the action specified in the entry. If it is F (forward), the next-best match is selected to receive the call. If no application is listening for this entry, then the next-best match is selected, and so on, until an entry that specifies R (reject) is selected and the call is rejected. A routing list entry with priority 1 is selected before one with priority 2, and so on.

What happens to a call whose characteristics do not match any routing list entry? The call is rejected with cause=0, diagnostic=0.

Understanding the X.25 Routing List

The routing list (/**etc/xrt.names**) is used to listen for incoming calls and route them to an application that can deal with them appropriately.

Managing the Routing List

Use the **xroute** command, to view, change, add to, and delete from the routing list. After you have configured the X.25 adapter, you may need to add entries to the list to meet additional requirements.

Called Address Extension

The *address extension* of the user for whom the call is intended. The maximum length is 40. An asterisk (*) in this field indicates that any characters are acceptable in the remainder of the address extension.

Priority 1, 2, or 3, where 1 has the highest priority. The priority is used in matching the details of an incoming call to routing list entries, to find a routing list entry for which an application is listening.

Action R for reject the incoming call, or F for forward the incoming call, when no application is listening for calls that fit the criteria specified in this entry.

Understanding xtalk Address Lists

An address list (the **xtalk.names** file) holds the names of people with whom you might want to communicate, together with their network user addresses and some other information. This saves you from having to know the address when you make a call. The address lists are used by the **xtalk** command.

There can be one system address list and one list for each user.

Directories

The system address list resides in /**etc**, and each user's own address list resides in the **$HOME** directory.

Managing the Address Lists

In addition to talking to other people, the **xtalk** command enables you to manage the X.25 address lists.

Fields

Each entry in an address list specifies the following details for one, named person:

Person's name
X.25 Port
Network user address
Address extension
Facilities

Person's Name The name can be up to 15 alphanumeric characters. You can have more than one name associated with an address; in this case you need one entry in the address file for each name.

Note that the other person referred to here may be another computer.

X.25 Port The name of the X.25 port to be used.

Network User Address The X.25 network user address of the other person, which can be up to 15 digits.

Address Extension The address extension can be up to 40 digits. This value is put into the address extension facility field in the call request packet.

Facilities A sequence of up to 16 bytes in hexadecimal format that can be put in the call-request packet for the **xtalk** command. These characters indicate the optional facilities that are required on calls to this network user address.

xtalk Overview

What xtalk Enables You to Do

The **xtalk** command lets you have conversations, by typing messages, over an X.25 network. You can also transfer files.

Making and Receiving Calls

Using the **xtalk** command has several features that are analogous to using the telephone. To have a conversation or transfer files, one of you must first make a call and the other must receive and accept the call. Either of you can end the call at any time.

Listening for Calls

To tell you when an incoming call arrives, you can start the **xtalk** command to listen for calls in the background and notify you when they arrive.

When a call arrives, you start the **xtalk** command again in the foreground and then choose whether to accept or reject the call.

Keeping an Address List

Each person you can talk to over an X.25 network has a network user address (similar to a telephone number). To help you make calls without knowing everybody's addresses, the **xtalk** command has a built-in address list. If you want to, you can use the system address list. Or you can keep your own local address list, which enables you to override the names in the system address list and add more names and addresses.

Having a Conversation

To have a conversation, you type messages; both your messages and the other person's messages appear on the panel on both screens.

You can record in a log file the messages you exchange during a conversation.

Transferring Files

To transfer files, one of you sends the file, and the other can choose to accept or reject it. You can choose whether to overwrite or append it to an existing file of the same name, or to save it under a new name.

You can send any ordinary file, but not a directory or special file.

Related Information

X.25 Commands Overview on page 15–36.
How to Make a Call with **xtalk** on page 15–65.
How to Get **xtalk** to Listen for Calls on page 15–63.
How to Receive a Call with **xtalk** on page 15–67.
How to Have an **xtalk** Conversation on page 15–68.
How to Log an **xtalk** Conversation on page 15–69.
How to Transfer Files with **xtalk** on page 15–71.
How to End a Call with **xtalk** on page 15–73.
How to Manage **xtalk** Address Lists on page 15–61.
X.25 Overview for Programming in *Communications Programming Concepts.*
The **xtalk**, **mail**, **talk**, and **ftp** commands.

Problem 2 Attempts to connect an X.25 port fail, whether using the **xmanage** command or an application using the **x25_link_connect** subroutine. The physical layer is established for several seconds, as shown by the lights on the NTU, but then goes down again. **xmonitor** shows that there is no line activity.

Suggestions

- The X.25 adapter is expecting a clock signal and not receiving one. Try to adjust the NTU to provide clocking.
- The cable is loose so that the clock pin is not connected.

Problem 3 Attempts to connect an X.25 port fail, whether using the **xmanage** command or an application using the **x25_link_connect** subroutine. The physical layer is connected but the frame layer fails to come up.

Suggestion

- The type of line attribute is DCE rather than DTE for this X.25 adapter. Use the **smit** command to change it to DTE.

Problem 4 The **xmanage** command connects the port, but the **xtalk** command cannot establish a session. The **xmonitor** command shows that SABMs are being sent and received and no restart is being sent.

Suggestion

- The connection mode attribute is set incorrectly. Use the **smit** command to change it.

Diagnosing Problems with Making an Outgoing X.25 Call

Problems and Suggestions

Problem 5 Starting X.25 communications for the very first time results in an **ENXIO** error.

Suggestion

- You must use the **smit** command to complete the initial configuration before using X.25 communications.

Problem 6 Incoming calls are arriving, but outgoing calls cannot be made, on a switched virtual circuit (SVC).

Suggestions

- The configuration of the SVC logical channel numbers is incorrect. Check your network subscription and use the **smit** command to check the network attributes.
- Use the **smit** command to check that outgoing calls and local charges are allowed.

Problem 11 When sending a data packet with the D-bit set on a permanent virtual circuit (PVC), a reset packet is sent instead.

Suggestion

- Use the **smit** command to configure the PVC to use D-bits.

Problem 12 When sending an interrupt packet with more than one byte of user data, a reset packet is sent instead.

Suggestion

- The 1980 version of X.25 supports exactly one byte of user data in the interrupt packet. The 1984 version supports up to 32 bytes. Check your subscription and the value of the CCITT support attribute in **smit**.

Problem 13 When sending large packets on slow lines, the link sometimes gets restarted.

Suggestion

- The CCITT timer, T1, may have expired. Use the **smit** command to increase the value of the T1 attribute, or use smaller packets.

Diagnosing X.25 Command Problems

Problems and Suggestions

Problem 14 Attempts to start up **xmonitor** twice on the same X.25 port fail.

Suggestion

- Only one process is allowed to access X.25 monitoring at one time.

Problem 15 The **xmanage** command does not allow monitoring to be started up for an X.25 port, even though it is showing that monitoring is not currently on.

Suggestion

- The **xmanage** command knows only about monitoring sessions that it starts. If the **xmonitor** command is started outside of **xmanage**, it will not be shown on the screen.

Problem 16 After starting a monitor session using the **xmanage** command, then quitting **xmanage** and restarting it, **xmanage** now shows that monitoring is off for that X.25 port.

Suggestion

- When it completes, the **xmanage** command stops any monitoring session that it had started.

Problem 17 None of the X.25 commands get past the title panel.

Suggestion

- Use **echo $TERM** to find out what the system thinks your terminal type is and check that it matches your terminal type.

X.25 Packet Switching Overview

These are some of the X.25 concepts with which you should be familiar if you intend to implement X.25 communications on the RISC System/6000.

Network communication with X.25
The three levels of X.25
X.25 packet switching
Network user addresses
Optional X.25 facilities
Logical channels and virtual circuits
X.25 packet types

Network Communication with X.25

X.25 is a communications protocol used in packet switching.

An X.25 network is similar to a telephone network, except that it conveys computer files and messages rather than speech. When you pick up a phone and dial a number, a path is set up through the telephone network (if the necessary lines are not busy). When the call is answered, the conversation can start and will last until you or the called person hangs up.

For all communication to work, a set of rules, or *protocol*, is needed. In a telephone call between two people, for instance, we tend to use standard, commonly understood phrases:

"Hello, Mary Smith speaking."
"Ah, this is Joe da Costa, please could I speak to Amanda."
"Yes, she's just here, I'll hand you over."
"Hello, Amanda here."
"Hello, this is Joe."

...until we know we have established a conversation and can start saying what we wanted to say. We end the call in a standard way too, so that both of us know when the conversation is over.

Given the necessary hardware and programs, one computer can be connected to another across the lines of a network, and they can exchange messages and files by sending and receiving data signals. Again, a protocol is needed, equivalent to that of a telephone conversation.

Several different protocols exist, but the only one that concerns us here is the one named X.25, which is designed for a form of communication known as *packet switching*.

X.25 simply refers to the 25th recommendation in the X series made by the International Telegraph and Telephone Consultation Committee (CCITT). Versions of the recommendation were agreed in 1976, 1980, 1984, and 1988. The International Organization for Standardization (ISO) has also published the X.25 recommendation as ISO 8208.

Differences Between Networks

The support provided by networks for X.25 communications varies. For instance, some support the 1980 version of the X.25 protocol and others the 1984 version. Some networks support some optional facilities but not others. Some networks allow the use of permanent virtual circuits (PVCs) while others do not. Different networks use different addressing standards. You must always check with your network provider that the features you want to use are available on your network.

This table shows the structure of the network user address (NUA), when 10 digits are allocated by the network provider to the NTN, leaving two digits for the subaddress:

Country Code	National Terminal Number	Optional Subaddress
123	4567890123	45

The X.25 Routing Overview describes how NUAs are used by X.25 communications on the RISC System/6000.

Optional X.25 Facilities

There are a number of *optional facilities* that the network provider may or may not provide, and to which your organization may or may not choose to subscribe. If you attempt to use one of these facilities on a line for which it is not being provided or subscribed to, the call may be cleared. The optional facilities for data terminal equipment (DTE) are specified by the CCITT and in ISO 8208 (reference); you have to specify which facilities you want to use, when you configure the X.25 adapter. The facilities supported on X.25 networks conforming to the 1980 or 1984 versions of X.25 are listed in Optional Facilities Control on page 15–11.

Logical Channels and Virtual Circuits

In communications terminology, the computer or workstation that sends and receives data is known as the *data terminal equipment (DTE).* The network equipment that is physically connected to the DTE is the *data circuit-terminating equipment* (DCE).

When one person makes a call to another, over the X.25 network, one of a predefined number of *logical channels* is assigned to the call. A logical channel is assigned at each end of the call, and each DTE includes a *logical channel number* in each packet sent. The number identifies the logical channel that connects the DTE with its DCE. The two logical channel numbers may be different, but each DTE needs to know only the number it assigned to the channel, and does not need to be aware of the other number.

When the two logical channels are assigned to a call, a *virtual circuit* is established from one DTE to the other, via the DCEs on the network. Each logical channel is either for outgoing calls only, incoming calls only, or two-way calls (or it is permanently connected), but once the virtual circuit is established it is always for two-way communication. The virtual circuit may be either switched or permanent.

X.25 Packet Types

Different types of packet are used for such purposes as making a call, accepting a call, transferring data, and terminating a call. The X.25 communications software does most of the work involved in creating the packets. You don't have to know the detailed content of each packet; you only have to supply the information that is needed to create the packet.

In some circumstances, the contents of the packet when it reaches the called DTE are different from when it left the calling DTE. This is because some information is different for each DTE (for example, logical channel number), or only applicable to one of the DTEs, or it is information inserted by the network.

More information is available for the packets listed here. (Only some of the fields in the packets are shown; if you want to see complete packet layouts, and read more detailed information, look at ISO 8208, reference in the X.25 Communications Bibliography on page 15–3.)

Call-request packet
Incoming-call packet
Call-accepted packet
Call-connected packet
Data packet
Clear-request packet
Clear-indication packet
Clear-confirmation packet
Interrupt packet
Interrupt-confirmation packet
Reset-request packet
Reset-indication packet
Reset-confirmation packet

(Other packet types are not described in detail in the RISC System/6000 information, but you can find out about them from ISO 8208, reference in the X.25 Communications Bibliography on page 15–3):

Receive-ready packet
Receive-not-ready packet
Reject packet
Restart-request packet
Restart-indication packet
Restart-confirmation packet
Diagnostic packet

Related Information

Network Overview on page 5–1.
X.25 Communications Overview for System Management on page 15–1.
X.25 Calls Overview: Packet Level on page 15–40.
X.25 Communications: Bibliography on page 15–3.

X.25 Commands Overview

The X.25 Commands and What They Do

The X.25 commands enable you to make use of the X.25 network without doing any application programming yourself.

xcomms Choose one of the other commands.

xtalk Communicate with other people.

Manage address lists for outgoing calls.

xroute Manage a routing list for incoming calls.

xmanage Display status information for an X.25 port

Connect and disconnect an X.25 port

Get statistics for an X.25 port .

xmonitor Monitor the activity on an X.25 port.

Before you can use the X.25 commands, X.25 communications must be installed and configured. If you intend to use the **xcomms**, **xtalk**, **xmanage**, and **xmonitor** commands, you must install the base operating system extension that includes them. (**xroute** is installed as part of the base operating system.)

Security Permissions Needed for the X.25 Commands

xcomms No permission needed.

xtalk For communicating, no permission needed.

For managing an address list, no permission needed.

xroute NET_CONFIG permission.

Selecting an Action

Within an application, the possible actions are displayed across the top of the screen. One of the actions is always highlighted and selected. To select an action, use the Left and Right keys. For some actions, you *also* need to make sure the selected object or option is the one you want, and press Enter.

Selecting an Object or Option

Some panels give you a vertical list of objects (such as entries in the address or routing list) or options (such as the list of applications that **xcomms** gives you). One of the objects or options is always highlighted and selected. To select an object or option, use the Up and Down keys. You *also* need to make sure the selected action is the one you want, and press Enter.

Using the Main Panels in xroute and xtalk

xroute is used for managing the routing list and **xtalk** for managing the address lists. Both display on a *main panel* some of the information from the list.

Displaying Further Entries

To display entries you cannot see currently, use the Up and Down keys or the Page Up and Page Down keys. As you move the cursor up or down the screen, the entry the cursor is on is selected (and highlighted), and when you reach the top or bottom, the entries move downwards or upwards.

Displaying More Details About an Entry

To display more details of a particular entry, select the entry; then select the `BROWSE` action and press Enter.

Moving Between Fields on the ADD and CHANGE Panels

When you have selected `ADD` or `CHANGE` in **xtalk** or **xroute**, you may want to change the details for the selected entry. To move the cursor from one field to another, so that you can add or change data, press the Tab key to go to the next field, and Shift-Tab to go to the previous field.

Typing and Editing Data

There are several places where you have to type some text in a fixed-length field on the screen; for instance, a network user address.

Sometimes you type something in, and then want to change it. You can move along the text using the Left and Right cursor keys. You can use the Insert, Delete, Backspace, Home, and End keys.

Pressing Enter either commits all the data you have entered, or takes you to a blank field (if filling in that field is mandatory).

Using Break Keys

If you press a Break key, the program exits immediately and you *lose all changes.*

Handling System Messages

Sometimes a message is displayed on a panel. After you have read it, you must remove the message before you can do anything else. To do this, press the Escape key (Esc).

Some messages cannot be displayed on a panel, but are instead printed directly to the standard error destination (**stderr**).

X.25 Calls Overview: Packet Level

Switched Virtual Circuits (SVCs)
Permanent Virtual Circuits (PVCs)

Switched Virtual Circuits (SVCs)

To help you understand the sequence of events that happen during a call between two DTEs using a switched virtual circuit (SVC), the diagram on page 15–41 shows a very simple call. The two DTEs are A and B:

1. A makes a call, which B receives.
2. B accepts the call; A receives a message saying the call has been connected.
3. A sends some data, but does not ask for acknowledgement.
4. A sends some more data, which B acknowledges.
5. A clears the call; B receives indication of this, and confirms that it has received the indication.
6. The network confirms to A that the call has been cleared.

After the call is established, until it is cleared, many more data packets could be sent, in either direction. Interrupt packets and reset packets could also be sent, as necessary.

These events and the packets associated with them are discussed in more detail in:

X.25 Packet Switching: Making and Receiving a Call
X.25 Packet Switching: Transferring and Acknowledging Data
X.25 Packet Switching: Clearing, Resetting, and Interrupting Calls

Related Information for Programming

The *Communications Programming Concepts and Procedures* book includes the following information:

Two example programs show how this simple call could be implemented using the X.25 application programming interface (API):

X.25 Example Program svcxmit: Make a Call Using an SVC
X.25 Example Program svcrcv: Receive a Call Using an SVC.

The subroutines and identifiers used by these programs are discussed in:

X.25 API: Initializing and Terminating
X.25 API: Using the Connection Identifier for Calls
X.25 API: Using Counters to Correlate Messages
X.25 API: Listening for Incoming Calls
X.25 API: Making and Receiving a Call
X.25 API: Transferring and Acknowledging Data
X.25 API: Clearing, Resetting, and Interrupting Calls.

Permanent Virtual Circuits (PVCs)

On a permanent virtual circuit (PVC), you do not have to make a call before sending data. Once a PVC has been configured (using the **smit** command), it can be allocated to an application program, which can then send and receive data. If resynchronization is necessary, the circuit can be reset.

These events and the packets associated with them are discussed in more detail in:

X.25 Packet Switching: Transferring and Acknowledging Data
X.25 Packet Switching: Clearing, Resetting, and Interrupting Calls.

X.25 Packet Switching: Making and Receiving a Call

The example call shows DTE A making a call to DTE B, which accepts it. (Note that on a permanent virtual circuit, you do not have to make and receive calls.)

Call-Request and Incoming-Call Packets

To start the call, A tells the X.25 network that it wants to make a call to B, by sending a *call-request* packet that contains the following information:

- Packet header
- Logical channel number
- Called address
- Calling address
- Optional facilities.

The packet includes the logical channel number that A will use to identify this connection in all packets it sends during the call.

The D-bit can be set in the call-request packet, to give permission for the D-bit to be set in data packets during the call. This requests acknowledgment of the data sent in each packet.

The call user data (CUD) is up to 16 bytes of data (128, for a fast-select call), which may, for example, indicate the call type. X.25 does not specify what the call user data should be; the contents must be agreed upon by applications.

The call-request packet is known as an *incoming-call* packet when it reaches B.

Call-Accepted and Call-Connected Packets

B is already listening for calls.

If B decides to accept the incoming-call packet from A, it sends a *call-accepted* packet to the network that contains the following information:

- Packet header
- Logical channel number
- Called address (optional)
- Calling address (optional)
- Optional facilities
- Called user data (fast-select calls only).

The call-accepted packet includes the logical channel number that B will use to identify this connection in all packets it sends during the call; this is usually a different number from that used by A for the same call.

The call accepted packet is known as a *call-connected* packet when it reaches A. The switched virtual circuit (SVC) connecting A and B is now established.

If you are using the X.25 application programming interface to do this, see:

X.25 API: Listening for Incoming Calls
X.25 API: Making and Receiving a Call.

In the example, A sends the clear-request packet. The network sends a *clear-indication* packet to B, telling it that the call has been cleared. When B receives the clear-indication packet, a clear-confirmation packet is sent to the DCE. The network sends a *clear-confirmation* packet back to A.

Interrupt and Interrupt-Confirmation Packets

The X.25 network delivers data packets in the same order as they are sent. However, there may be a need to send data that is independent of the normal data flow, for example, a break key. To do this, either DTE can send an *interrupt* packet.

An interrupt packet can contain one byte of data on networks supporting the 1980 version of X.25, or up to 32 bytes on networks supporting the 1984 version.

The X.25 network transmits an interrupt packet as quickly as possible, bypassing the flow of data packets if necessary. The *interrupt-confirmation* packet confirms that the other DTE has read the interrupt packet. An interrupt-confirmation packet must be received before the network will accept another interrupt packet. Sending an interrupt packet is less drastic than resetting the call. It should be followed by whatever procedure has been agreed upon to re-establish the flow of communications.

One application may want to send an interrupt packet if, for instance, it has received no acknowledgement of data packets sent, and wants to find out what happened to them before sending more. Another use of interrupts is for Open System Interconnection (OSI) expedited data transfer.

Reset-Request, Reset-Indication, and Reset-Confirmation Packets

A virtual circuit can be reset by either DTE to discard any data in transit and reinitialize communications. Typically, this is done after an error condition is detected by a DTE.

To reset a virtual circuit, a DTE sends a *reset-request* packet. This is received as a *reset-indication* packet, and it may remove all the data and interrupt packets already on the virtual circuit. A *reset-confirmation* packet must be received before the network will accept any more data or interrupt packets.

If you are using the X.25 application programming interface to do this, see:

X.25 API: Clearing, Resetting, and Interrupting Calls in *Communications Programming Concepts*.

X.25 Clear and Reset Codes Overview

Each clear-indication or reset-indication packet includes a 1-byte cause code and a 1-byte diagnostic code included in the received data. The API subroutines take no specific action on any of the cause or diagnostic codes.

Where the Clear or Reset Came From

X.25 clear-indication and reset-indication packets may be generated by the X.25 adapter code, the remote data terminal equipment (DTE), or the X.25 network itself. The relationship between cause codes and diagnostic codes is shown in a table in the annexes to ISO 8208 (reference).

When SNA services are being used, use the List of SNA Diagnostic Codes; otherwise use the List of CCITT/ISO Diagnostic Codes. Some additional diagnostic codes are generated by the **xtalk** application (see Diagnostic Codes Used by **xtalk**, on page 15–50).

Related Information

X.25 Problem Diagnosis Overview on page 15–25.
List of X.25 Diagnostic Codes on page 15–48.
List of X.25 Clear and Reset Cause Codes on page 15–47.
List of X.25 Logical Channel States on page 15–54.
X.25 Packet Switching Overview on page 15–30.
X.25 Overview for System Management on page 15–1.

List of X.25 Diagnostic Codes

Diagnostic codes give additional information about the reason for sending a clear-indication or reset-indication. (The reason is also indicated in the cause code.) The meaning of each diagnostic code depends on whether X.25 is being used as a medium for SNA communications, via qualified logical link control (QLLC) or being used directly. If SNA is being used, refer to the List of SNA Diagnostic Codes; if X.25 is being used directly, refer to the List of CCITT/ISO Diagnostic Codes.

In addition, some diagnostic codes are used by the **xtalk** command.

List of CCITT/ISO Diagnostic Codes

The following diagnostic codes are set in clear- and reset-indication packets, when SNA services are *not* being used. The states refer to the logical channel states.

Hex	Dec	Meaning
00	0	Clear or reset generated during restart
01	1	Invalid P(S) in packet from DCE
02	2	Invalid P(R) in packet from DCE
10	16	Invalid packet type
11	17	Invalid packet from DCE in state r1
12	18	Invalid packet from DCE in state r2
13	19	Invalid packet from DCE in state r3
14	20	Invalid packet from DCE in state p1
15	21	Invalid packet from DCE in state p2
16	22	Invalid packet from DCE in state p3
17	23	Invalid packet from DCE in state p4
18	24	Invalid packet from DCE in state p5
19	25	Invalid packet from DCE in state p6
1A	26	Invalid packet from DCE in state p7
1B	27	Invalid packet from DCE in state d1
1C	28	Invalid packet from DCE in state d2
1D	29	Invalid packet from DCE in state d3
20	32	Packet not allowed
21	33	Unidentifiable packet received from DCE
22	34	Incoming call received on one-way channel
23	35	Clear or call packet received from DCE on a permanent virtual circuit (PVC)
24	36	Packet received on an unassigned logical channel
25	37	REJECT not subscribed to
26	38	Packet received from DCE was too short
27	39	Packet received from DCE was too long
28	0	Invalid general format identifier (GFI)
29	41	Restart packet received from DCE with non-zero logical channel identifier
2A	42	Invalid fast-select packet received from DCE
2B	43	Unauthorized interrupt confirmation
2C	44	Interrupt packet received from DCE when acknowledgment was still outstanding
2D	45	Unauthorized reject
30	48	Timer expired (or limit surpassed)
31	49	Time-out or retries reached on call response from DCE
32	50	Time-out or retries reached on clear response from DCE
33	51	Time-out or retries reached on reset response from DCE

List of Diagnostic Codes Used by xtalk

The following diagnostic codes are set up by the X.25 supplied application, **xtalk**, when clearing connections:

Hex	Dec	Meaning
F1	241	Normal disconnection
F4	244	Connection request rejected. This may be because the program is busy (already connected to someone else), or because the other program is not listening.

Related Information

X.25 Communications: Bibliography on page 15–3.
X.25 Problem Diagnosis Overview on page 15–25.
List of X.25 Clear and Reset Cause Codes on page 15–47.
X.25 Packet Switching: Clearing, Resetting, and Interrupting Calls on page 15–51.
X.25 Clear and Reset Codes Overview on page 15–46.

Hex	Dec	Meaning
56	86	QLLC error: Frame reject received
57	87	QLLC error: Header invalid
58	88	QLLC error: Data received in wrong state
59	89	QLLC error: Time-out condition
5A	90	QLLC error: Nr invalid
5B	91	QLLC error: Recovery rejected or terminated
5D	93	QLLC error: ELLC time-out condition
60	96	PSH error (general)
61	97	PSH error: sequence error
62	98	PSH error: header too short
63	99	PSH error: PSH format invalid
64	100	PSH error: command undefined
65	101	PSH error: protocol invalid
66	102	PSH error: data received in wrong state
69	105	PSH error: time-out condition
70	112	PAD error (general)
71	113	PAD error: PAD access facility failure
72	114	PAD error: SDLC FCS error
73	115	PAD error: SDLC time-out
74	116	PAD error: SDLC frame invalid
75	117	PAD error: I-field too long
76	118	PAD error: SDLC sequence error
77	119	PAD error: SDLC frame aborted
78	120	PAD error: SDLC FRMR received
79	121	PAD error: SDLC response invalid
7B	123	PAD error: invalid packet type
7F	127	PAD error: PAD inoperable
80	128	DTE-specific (general)
81	129	DTE-specific: 8100_DPPX-specific
82	130	DTE-specific: INN_QLLC-specific
83	131	DTE-specific: INN_QLLC-specific
84	132	DTE-specific: INN_QLLC-specific
85	133	DTE-specific: INN_QLLC-specific
86	134	DTE-specific: INN_QLLC-specific
87	135	DTE-specific: INN_QLLC-specific
88	136	DTE-specific: INN_QLLC-specific
89	137	DTE-specific: INN_QLLC-specific
8A	138	DTE-specific: INN_QLLC-specific
8B	139	DTE-specific: INN_QLLC-specific
8C	140	DTE-specific: INN_QLLC-specific
8D	141	DTE-specific: INN_QLLC-specific
8E	142	DTE-specific: INN_QLLC-specific
8F	143	DTE-specific: INN_QLLC-specific
90	144	Network-specific
91	145	Network-specific: DDX–P RNR packet received
A0	160	Packet not allowed (general)
A1	161	Packet not allowed: invalid M-bit packet sequence
A2	162	Packet not allowed: invalid packet type received
A3	163	Packet not allowed: invalid packet on PVC
A4	164	Packet not allowed: unassigned LC
A5	165	Packet not allowed: diagnostic packet received
A6	166	Packet not allowed: packet too short
A7	167	Packet not allowed: packet too long

List of X.25 Logical Channel States

This article lists the CCITT logical channel states referred to in the List of X.25 Diagnostic Codes.

Logical Channel States

State	**Meaning**
d1	Flow control ready
d2	DTE reset request
d3	DCE reset indication
p1	Channel ready
p2	DTE call request
p3	DCE incoming call
p4	**Data transfer**
p5	Call collision
p6	DTE clear request
p7	DCE clear indication
r1	Packet level ready
r2	DTE restart request
r3	DCE restart indication.

Related Information

List of X.25 Diagnostic Codes on page 15–48.
X.25 Communications: Bibliography on page 15–3.
Logical Channels and Virtual Circuits on page 15–32.

How to Revert to Initial Configuration Defaults

Some important X.25 adapter attributes have default values that can be set by the initial configuration process, based on the network identifier and country code. If you ever want to reset these attributes to these default values, without affecting other attributes you have set:

1. Use **smit** to do a complete initial configuration.

How to Revert to Unconfigured Default Values for All Attributes

If you ever want to return to the default values for *all* attributes, losing all the changes you have made:

1. Remove the device definition from the Customized Devices Object Class, using the **rmdev** command with the **–d** flag. For example:

   ```
   rmdev –d –l x25s0
   ```

2. Make a new device definition in the Customized Devices Object Class, using the **mkdev** command. For example:

   ```
   mkdev –c adapter –s mca –t x25 –p bus0 –w 0
   ```

   The **–w** flag specifies the slot number which, in this example is 0.

3. Use **smit** to do a complete initial configuration.

How to Change to a New Network

If you change from one network to another, first decide whether to set all the attributes back to the original default values, or preserve the modifications you made to the attributes. Your decision should be based on how similar your setup is on the old and new networks. If you decide to retain your attribute settings:

1. Revert to initial configuration defaults

Otherwise:

1. Revert to unconfigured default values for all attributes.

You may also have to reconfigure SNA or TCP/IP.

How to Set Up Permanent Virtual Circuits

If you are using permanent virtual circuits, you need to do the following:

1. Follow the procedure for How to Change X.25 Adapter Attributes.
2. At step 4, select

   ```
   Change / Show Network Attributes
   ```

3. Enter the number of PVCs you have subscribed to, and the lowest logical channel number.
4. Commit your changes.
5. Select

   ```
   Change / Show Default for Permanent Virtual Circuits (PVC)
   ```

6. Check the supplied default values against the details of your network subscription. Change the values as appropriate.
7. Commit your changes.
8. If you need to specify different values for individual PVCs, select

   ```
   Change / Show a Specific Permanent Virtual Circuit (PVC)
   ```

9. Enter the PVC identifier.
10. Type in the PVC logical channel number and any values that need to differ from the default PVC you have defined.
11. Commit your changes.

Related Information

X.25 Configuration Attributes Overview on page 15–7.
List of X.25 Network Configuration Attributes on page 15–8.
List of X.25 Packet Configuration Attributes on page 15–10.
List of X.25 PVC Configuration Attributes on page 15–19.
List of X.25 Frame Configuration Attributes on page on page 15–16.
List of X.25 General Configuration Attributes on page 15–18.
X.25 Packet Switching Overview on page 15–30.
Network User Addresses on page 15–31.
The SMIT Overview in *General Concepts and Procedures.*
Using SMIT in *General Concepts and Procedures.*

For SNA:

Defining X.25 Attachment Characteristics on page 13–149.
Defining QLLC (X.25) Logical Link Characteristics on page 13–157.
Defining X.25 Physical Link Characteristics on page 13–162.

For TCP/IP:

How to Configure a Network Interface on page 14–100.
X.25 Communications Overview for System Management on page 15–1.
Configuring X.25 on page 15–5 .

How to Change an Entry in the Routing List

1. Start **xroute**.
2. You can change all the information in the entries on the list. Using the Up and Down keys, select the entry you want to change. Using the Left and Right keys, select the CHANGE action and press Enter.
3. The current information is displayed on a panel similar to the BROWSE panel. To change the information, type over it.

xroute prevents you from changing any entry that is currently being used by an application to listen for calls.

How to Delete an Entry from the Routing List

1. Start **xroute**.
2. Using the Up and Down keys, select the entry you want to delete. Using the Left and Right keys, select the DELETE action and press Enter.
3. **xroute** displays the entry on a panel similar to the BROWSE panel.
4. Press Enter again to confirm that you want the entry removed. To abandon the deletion, press the Escape key (Esc).

xroute prevents you from deleting any entry that is currently being used by an application to listen for calls.

Related Information

How to Manage **xtalk** Address Lists on page 15–61.
X.25 Commands Overview on page 15–36.
Diagnosing X.25 Command Problems on page 15–28.
Diagnosing Problems with Receiving an Incoming X.25 Call on page 15–27.
The **xroute** command.
X.25 Routing Overview on page 15–20.
Understanding the X.25 Routing List on page 15–20.

How to Add an Entry to the Address Lists

1. Start **xtalk**.
2. Before adding a new name, find out the details, such as network user address, from the person whose name you want to add.
3. Using the Left and Right keys, select the ADD action. If you do not have write access to the system address list, any new entries will be added to the local list. If you do have write access to the system address list, **xtalk** asks you if you want to add a name to the local list or the system list. Select the appropriate list and press Enter.
4. **xtalk** then displays a panel, on which you should fill in the appropriate information and press Enter.

When adding to your own list, **xtalk** does not check whether the name is used in the system list (or in any other user's list). You can, therefore, override the names in the system list with your own names.

You can add (to one or both lists) more than one name for the same address.

How to Change an Entry in the Address Lists

1. Start **xtalk**.
2. You can change all the information in the entries on the lists. Using the Up and Down keys, select the entry you want to change. Using the Left and Right keys, select the CHANGE action and press Enter.
3. The current information is displayed on a panel similar to the BROWSE panel.
4. To change the information, type over it.

How to Delete an Entry from the Address Lists

1. Start **xtalk**.
2. Using the Up and Down keys, select the entry you want to delete. Using the Left and Right keys, select the DELETE action and press Enter.
3. Press Enter again to confirm that you want the entry removed.

Related Information

How to Manage the X.25 Routing List on page 15–59.
X.25 Commands Overview on page 15–36.
Diagnosing X.25 Command Problems on page 15–28.
The **xtalk** command.
X.25 Routing Overview on page 15–20.

How to Start Listening for Calls in the Background

1. Start the **xtalk** command with the **–l** flag and the **–n** flag. The **–l** flag specifies the name of the entry in the routing list that you want to use. The **–n** flag tells the **xtalk** command to run as a background process, and it will do so until you kill it. To use an entry named **mycall,** enter:

   ```
   xtalk –n –l mycall
   ```

   To use an IBM-supplied routing name suitable for receiving calls from other people using **xtalk**, enter:

   ```
   xtalk –n –l IBMXTALK
   ```

   You should put the **xtalk –n** command in your **$HOME/.profile**, so that listening starts as soon as you log in.

Related Information

How to Make a Call with **xtalk** on page 15–65.
How to Receive a Call with **xtalk** on page 15–67.
How to Have an **xtalk** Conversation on page 15–68.
How to Log an **xtalk** Conversation on page 15–69.
How to Transfer Files with **xtalk** on page 15–71.
How to End a Call with **xtalk** on page 15–73.
Understanding the X.25 Routing List on page 15–20.
How to Manage the X.25 Routing List on page 15–59.
Diagnosing Problems with Receiving an Incoming X.25 Call on page 15–27.
Diagnosing X.25 Command Problems on page 15–28.
The **xtalk** and **xroute** commands.
xtalk Overview on page15–24 .
X.25 Routing Overview on page 15–20.

Related Information

How to Get **xtalk** to Listen for Calls on page 15–63.
How to Receive a Call with **xtalk** on page 15–67.
How to Have an **xtalk** Conversation on page 15–68.
How to Log an **xtalk** Conversation on page 15–69.
How to Transfer Files with **xtalk** on page 15–71.
How to End a Call with **xtalk** on page 15–73.
How to Manage **xtalk** Address Lists on page 15–61.
Diagnosing X.25 Command Problems on page 15–28.
The **xtalk** command.
xtalk Overview on page 15–24.
X.25 Routing Overview on page 15–20.

How to Have an xtalk Conversation

Prerequisite Tasks

1. Set up X.25 communications. The base operating system extension that includes the **xtalk** command must be installed.
2. Before having a conversation, you must make or receive a call.

Procedure

When the called person accepts the call, **xtalk** displays the commands panel. This panel is displayed on your screen and the other person's screen. The commands panel offers you the following options:

```
TRANSFER FILE
BEGIN LOGGING
END LOGGING
CHANGE LOG FILENAME
QUIT CALL
```

This gives you the opportunity to begin logging your messages before you start the conversation.

If all you want to do is type messages to the other user, press the F2 key to display the messages panel. You can now type messages in the large blank space. Messages from the other person are displayed in the same space on the screen.

You can type text and edit it only one line at a time. When you press the Enter key, the message is transmitted. Incoming messages are displayed on the same panel. You need not wait for a response from the other user before typing another message yourself.

To display the commands panel again, press the F2 key. Whenever the commands panel is displayed, incoming messages are queued until you switch back to the messages panel.

Related Information

How to Make a Call with **xtalk** on page 15–65.
How to Get **xtalk** to Listen for Calls on page 15–63.
How to Receive a Call with **xtalk** on page 15–67.
How to Log an **xtalk** Conversation on page15–69.
How to Transfer Files with **xtalk** on page 15–71.
How to End a Call with **xtalk** on page 15–73.
Diagnosing X.25 Command Problems on page 15–28.
The **xtalk** command.
xtalk Overview on page 15–24.
X.25 Routing Overview on page 15–20.

How to Change the Name of the xtalk Log File

1. If necessary, press the F2 key to display the commands panel.
2. Using the Up and Down keys, select `Change log file name` and press the Enter key.
3. **xtalk** displays a panel on which you can type the new file name. The old log file is closed before the new file is started. The new log file name is used during the current call only.

How to Stop xtalk Logging

1. If necessary, press the F2 key to display the commands panel.
2. Using the Up and Down keys, select `END LOGGING` and press the Enter key.
3. **xtalk** closes the log file.

Related Information

How to Make a Call with **xtalk** on page 15–65.
How to Get **xtalk** to Listen for Calls on page 15–63.
How to Receive a Call with **xtalk** on page 15–67.
How to Log an **xtalk** Conversation on page15–69.
How to Transfer Files with **xtalk** on page 15–71.
How to End a Call with **xtalk** on page 15–73.
Diagnosing X.25 Command Problems on page 15–28.
The **xtalk** command.
xtalk Overview on page 15–24.
X.25 Routing Overview on page 15–20.

Related Information

How to Make a Call with **xtalk** on page 15–65.
How to Get **xtalk** to Listen for Calls on page 15–63.
How to Receive a Call with **xtalk** on page 15–67.
How to Log an **xtalk** Conversation on page15–69.
How to Transfer Files with **xtalk** on page 15–71.
How to End a Call with **xtalk** on page 15–73.
Diagnosing X.25 Command Problems on page 15–28.
The **xtalk** command.
xtalk Overview on page 15–24.
X.25 Routing Overview on page 15–20.

How to Display Status Information for an X.25 Port

Prerequisite Tasks

1. Set up X.25 communications. Install the base operating system extension that includes the **xmanage** command.
2. The information is refreshed at the interval specified by the **XMG_REFRESH** environment variable. The default is 5 seconds, allowing you to leave **xmanage** running to give up-to-date information about each X.25 port.

   If you do not want the information to be refreshed, set **XMG_REFRESH** to 0.

Procedure

1. Following the shell prompt, enter:

   ```
   xmanage
   ```

   Or:

1. Following the shell prompt, enter:

   ```
   xcomms
   ```

2. Select **xmanage** from the **xcomms** menu.

The **xmanage** command displays a list of ports with the following information about each:

Port name The name of the X.25 port, in the form x25s*n*, where *n* is single digit. The name of the X.25 port is the name assigned to the adapter by the installation process. Use the **smit** command to find out which adapter names can be used. This is the name that must be specified as the **link_name** parameter on any of the X.25 API subroutines that need it. It is used to identify the X.25 adapter.

Port status The status of each of the three levels of X.25 (physical, frame, and packet levels) is shown as `connecting`, `connected`, or `disconnected`.

Monitoring status Shows whether or not the X.25 port is currently being monitored (monitoring status is either `on` or `off`).

From the **xmanage** panel, you can connect or disconnect the ports, get statistics, or turn monitoring on or off.

How to Connect and Disconnect an X.25 Port

Prerequisite Tasks

1. Set up X.25 communications. You must install the base operating system extension that includes the **xmanage** command.
2. To use **xmanage** to connect or disconnect an X.25 port, you must have NET_CONFIG permission.

How to Connect an X.25 Port

1. Use the **xmanage** command to display a list of X.25 ports.
2. Using the Up and Down keys, select the X.25 port. Using the Left and Right keys, select the `CHANGE STATUS` action and press Enter.
3. Using the Up and Down keys, select the `CONNECT` option, and press Enter.

How to Disconnect an X.25 Port

1. Use the **xmanage** command to display a list of X.25 ports.
2. Using the Up and Down keys, select the X.25 port. Using the Left and Right keys, select the `CHANGE STATUS` action and press Enter.
3. Using the Up and Down keys, select the `DISCONNECT` option, and press Enter. If there are calls in progress, **xmanage** prompts you either to confirm that you want to disconnect the port or to abandon the action. **xmanage** displays a message while the port is being disconnected.

Related Information

How to Monitor an X.25 Port on page 15–77, How to Get Statistics for an X.25 Port on page 15–79, How to Connect and Disconnect an X.25 Port on page 15–76, How to Configure the X.25 Adapter on page 15–55, Diagnosing Problems with Connecting to the X.25 Network on page 15–25. X.25 Ports and Links Overview on page 15–35.

X.25 Commands Overview on page 15–36, Diagnosing X.25 Command Problems on page 15–28.

The SMIT Overview in *General Concepts and Procedures*, Using SMIT in *General Concepts and Procedures.*

The **xmanage** and **xmonitor** commands.

The **x25_link_connect**, **x25_link_disconnect**, **x25_link_query**, **x25_link_monitor** and **x25_link_statistics** subroutines.

How to End X.25 Monitoring

If you used **xmanage** to start monitoring, you can use it to end monitoring too. On the **MONITOR** panel, use the Up and Down keys to select the `END MONITORING` option, and press the Enter key.

If you used **xmonitor** to start monitoring, you must end it by terminating the process.

Related Information

How to Connect and Disconnect an X.25 Port on page 15–76, How to Get Statistics for an X.25 Port on page 15–79, X.25 Commands Overview on page 15–36, Diagnosing X.25 Command Problems on page 15–28, X.25 Ports and Links Overview on page 15–35.

The **xmanage** and **xmonitor** commands.

The **x25_link_monitor** API subroutine.

Example of X.25 Statistics

This is an example of the statistics output produced by the **STATISTICS** action on the **xmanage** panel, with an explanation of the terminology.

```
Statistics for X.25 port x25s0 are

   Active VC:  00020

   Bytes   Tx: 13056 Rx: 11392
   Packets Tx: 00102 Rx: 10721
   Errors  Tx: 00007 Rx: 00089

   Frame Statistics
   T1 Expirations: 00005  Level 2 conn: 00002  Level 2 disc: 00000
   T4 Expirations: 00003  Carrier loss: 00003  Connect time: 00513

   Packet Statistics
   Data        Packets Tx : 97821  Rx: 88773
   Restart     Packets Tx : 00003  Rx: 00000
   Restart–C   Packets Tx : 00000  Rx: 00003
   Diagnostic Packets Tx : 00000  Rx: 00000
   Packet Tx Error        : 00000  VC Establishments     : 00023
   Packet Timer Expirations :
       T20 : 00005   T21 : 00003   T22 : 00001   T23 : 00004
       T24 : 00010   T25 : 00002   T26 : 00006   T28 : 00000
```

Active VC The number of virtual circuits currently established.

Tx Transmitted.

Rx Received.

Level 2 conn Number of times the frame level has been connected.

Level 2 disc Number of times the frame level has been disconnected.

Carrier loss Number of times the carrier signal has been lost.

Restart_C Restart-confirmation.

VC Establishments The number of virtual circuits that have been established for this port.

Related Information

How to Get Statistics for an X.25 Port on page 15–79, How to Monitor an X.25 Port on page 15–77, How to Connect and Disconnect an X.25 Port on page 15–76, ,X.25 Packet Types on page15–34, X.25 Ports and Links Overview on page 15–35.

The **xmanage** command.

The **x25_link_statistics** subroutine.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Department 997, Building 997
11400 Burnet Rd.
Austin, Texas 78758–3493

Fold

Fold

Cut or Fold Along Line

Fold and Tape

Please Do Not Staple

Fold and Tape

International Business Machines
Corporation
11400 Burnet Road
Austin, Texas 78758–3493

Printed in the
United States of America

SC23-2203-00

SC23-2203-00