IBM Communications Server for Data Center Deployment
on AIX or Linux

IBM

APPC Application Suite

Version 7.0

IBM Communications Server for Data Center Deployment
on AIX or Linux

APPC Application Suite

Version 7.0

Sixth Edition (December 2012)

This edition applies to IBM Communications Server for Data Center Deployment on AIX or Linux, Version 7.0, program number 5725-H32, and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

IBM welcomes your comments. You may send your comments to the following address.
International Business Machines Corporation
Attn: z/OS Communications Server Information Development
Department AKCA, Building 501
P.O. Box 12195, 3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195

You can send us comments electronically by using one of the following methods:

Fax (USA and Canada):
1+919-254-1258

Send the fax to "Attn: z/OS Communications Server Information Development"

Internet email:
comsvrcf@us.ibm.com

World Wide Web:
http://www.ibm.com/systems/z/os/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number. Make sure to include the following in your comment or note:

- Title and order number of this document

- Page number or topic related to your comment

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

Contents

Tables

v

Figures

About This Book

This book provides configuration and usage information for the IBM®
Communications Server for Data Center Deployment on AIX® or Linux APPC
Application Suite. It also explains how to install the **asuite.tps** file, which is used
to select and execute the correct partner program, by using the **snatpinstall**
program.

The applications described in this book consist of pairs of TPs (transaction
programs) that run on an SNA network. Each application uses independent Logical
Unit (LU) 6.2 protocols to communicate with partner applications running on other
nodes in an SNA network. The applications can be accessed from a server or from
a Remote API Client. For a list of applications, see "What Is APPC Application
Suite?" on page 1.

This manual applies to IBM Communications Server for Data Center Deployment
(Communications Server), program product number 5725-H32, which is an IBM
software product that enables a server running AIX, or a computer running Linux,
to exchange information with other nodes on an SNA network.

There are three different installation variants of IBM Communications Server for
Data Center Deployment, depending on the hardware on which it operates:

IBM Communications Server for Data Center Deployment on AIX (CS/AIX)
> IBM Communications Server for Data Center Deployment on AIX operates
> on a server running AIX Version Version 6.1 or 7.1 base operating system.

IBM Communications Server for Data Center Deployment on Linux (CS Linux)
> IBM Communications Server for Data Center Deployment on Linux
> operates on the following:
> - 32–bit Intel workstations running Linux (i686)
> - 64–bit AMD64/Intel EM64T workstations running Linux (x86_64)
> - IBM pSeries® computers running Linux (ppc64)

**IBM Communications Server for Data Center Deployment on Linux for System
z® (CS Linux for System z)**
> IBM Communications Server for Data Center Deployment on Linux for
> System z operates on System z mainframes running Linux for System z
> (s390x).

In this book, the name Communications Server is used to indicate any of these
variants, and the term "Communications Server computer" is used to indicate any
type of computer running Communications Server, except where differences are
described explicitly.

This book applies to Version 7.0 of Communications Server.

Who Should Use This Book

This book is intended for System Administrators and anyone using APPC
Application Suite.

System Administrators
> System Administrators install Communications Server, migrate it from an

earlier release, then configure the system for network connection, and maintain the system. System Administrators should be familiar with the Communications Server system and with the hardware on which Communications Server runs. They must also be knowledgeable about the network to which the system is connected and understand SNA concepts.

End Users

End users includes anyone who uses the APPC Application Suite for Communications Server programs and SNA communications software.

How to Use This Book

This section explains how information is organized and presented in this book.

Organization of This Book

This book is organized as follows:

- Chapter 1, "Overview," on page 1, provides an overview of APPC (Advanced Program-to-Program Communication) Application Suite for Communications Server and explains how to configure your system to run APPC applications.
- Chapter 2, "AFTP," on page 7, describes the **AFTP** application, provides configuration information, and explains how to transfer files. It also lists and describes the subcommands you can use within the **AFTP** shell.
- Chapter 3, "ACOPY," on page 19, describes the **ACOPY** application. It also provides configuration information and explains the command options.
- Chapter 4, "ANAME," on page 23, describes the **ANAME** application, provides configuration information, and explains how to add, remove, and view **ANAME** database records. It also summarizes the supported **ANAME** commands.
- Chapter 5, "APING," on page 31, describes the **APING** application, and summarizes the **APING** parameters and options.
- Chapter 6, "AREXEC," on page 35, describes the **AREXEC** application, provides configuration information, and summarizes the **AREXEC** parameters and options.
- Chapter 7, "ATELL," on page 37, describes the **ATELL** application, provides configuration information, and summarizes the **ATELL** parameters and options.

Typographic Conventions

Table 1 shows the typographic styles used in this document.

Table 1. Typographic Conventions

Special Element	Sample of Typography
Document title	*IBM Communications Server for Data Center Deployment on AIX or Linux APPC Programmer's Guide*
File or path name	**/etc/passwd**
Program or application	**ACOPY**
Command or AIX / Linux utility	**snaadmin define_userid_password**
Option or flag	**binary**
Parameter or Motif field	*partner LU name*
Literal value or selection that the user can enter (including default values)	VBA
Constant or signal	#BATCH
Variable representing a supplied value	*RemoteSystemID*
User input	**aping NETID.LUNAME7**

Table 1. Typographic Conventions (continued)

Special Element	Sample of Typography
Computer output	**Goodbye**
Function, call, or entry point	Send
Keyboard keys	**Return**

What Is New for This Release

Communications Server for Data Center Deployment Version 7.0 is a follow-on product to Distributed Communications Server Version 6.4, which continues to be supported.

Where to Find More Information

See the Bibliography for other books in the Communications Server library, as well as books that contain additional information about topics related to SNA and AIX or Linux workstations.

Chapter 1. Overview

This chapter provides an overview of the applications provided in the APPC Application Suite. You use these application programs to send and receive files, simplify network naming, check connectivity, execute commands on remote workstations, and send messages from one workstation to another. This chapter also explains how to configure your system to run these applications.

What Is APPC Application Suite?

APPC Application Suite is a set of command-line programs that communicate over APPC sessions. They use independent LU 6.2 protocols to communicate with similar applications running on other nodes in SNA networks. Each application comprises two programs: a requester program and a provider program. The applications can be accessed from a server or from a Remote API Client.

APPC Application Suite contains the following applications:
- **ACOPY** (APPC COPY)
- **AFTP** (APPC File Transfer Protocol)
- **ANAME** (APPC Name Server)
- **APING** (APPC Ping)
- **AREXEC** (APPC Remote EXECution)
- **ATELL** (APPC TELL)

The rest of this chapter describes how to configure your system to run APPC Application Suite. The subsequent chapters describe the programs in detail.

Configuring Your System for APPC Application Suite

You must establish links with any remote systems before running APPC Application Suite programs that communicate with those systems.

Configuration Overview

APPC Application Suite includes both requester programs and provider programs. Requester and provider programs are paired; the requester program requests services, and the provider program performs the services. A requester program resides on the local node or on a client connected to the local node. It requests services from a provider program residing on a partner node or on a client connected to the partner node. APPC Application Suite may be used both to access remote providers and to provide services to remote requesters.

For example, **AFTP** is a requester program that enables you to send and receive files between two computers; its provider program is **AFTPD**. Assume that **AFTP** resides on local node A and **AFTPD** resides on remote node B. To send a file from node A to node B, you invoke **AFTP**, which then communicates with the provider program **AFTPD** at node B, requesting that it execute the required task.

In order for a requester program to communicate with its provider program, you must identify the target LU and specify an SNA mode. You can supply this information on the command line or include it in a configuration file.

When specifying a target LU, the *destination name* can either be a *fully qualified LU name*, a *CPI-C* (Common Programming Interface for Communications) *symbolic destination name*, or an **ANAME** *alias*. If an **ANAME** *alias* conflicts with a *CPI-C symbolic destination name*, the application uses the *CPI-C symbolic destination name*. (For more information, see "Destination Names" on page 3.)

Configuring Your System

You must configure the local SNA node and its clients (if any) to support APPC Application Suite programs. Security and session configuration is optional depending on your needs. This section describes how to configure each of the following elements:

- Node
- Link
- Local LU
- Security
- Requester programs
- Provider programs

Configuring the Node

Configure the local node by providing a control point name for the node and the name of the local network to which the node belongs.

Configuring the Link

If the local node is:

- A low-entry networking (LEN) node, configure links to any other nodes that are required. You must also configure the partner LUs explicitly.
- An APPN end node (EN), configure a link to an APPN network node that acts as a network node server for the local node.
- An APPN network node (NN), configure links to other network nodes and any attached client end nodes.

Configuring the Local LU

Normally, APPC Application Suite applications can use the default LU associated with the local node (the control point LU), so you do not need to define any local LUs.

If you need to use different LUs, you should define each local LU that will be used for these applications (refer to the IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide for more information). The application user can then use the APPCLLU environment variable to specify which local LU to use. For more information on APPCLLU, refer to the IBM Communications Server for Data Center Deployment on AIX or Linux Administration Command Reference or the IBM Communications Server for Data Center Deployment on AIX or Linux CPI-C Programmer's Guide.

Configuring Security

As delivered with Communications Server, the APPC Application Suite provider programs do not require conversation security, so the user of a requester program does not need to specify a user ID and password. If you want to set up the provider programs to require conversation security, you will need to do the following:

- Define conversation security user IDs and passwords for the users who are permitted to access the provider programs. For the **AFTP** provider program

running on the local node or on a AIX / Linux client, the conversation security user IDs should match AIX / Linux login IDs on the local system. This ensures that the remote user, logging in to the **AFTP** provider with a specific user ID, can access only the files and directories that are available to the corresponding AIX / Linux login ID. On Windows clients, file access using **AFTP** is controlled by the **aftp.ini** file; see "AFTP Configuration" on page 7 for more information.

- Optionally, define a security access list to restrict the use of each provider program to a specific set of users.
- Define the target TP for each provider program to require conversation security (and optionally specify the security access list).

For more information, refer to the IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide.

If the provider programs on the remote system are set up to require conversation security, the user of a requester program on the local system must specify a user ID and password. For the **AFTP** application, these parameters are specified while using **AFTP** to log in to the remote system; for all other APPC Application Suite applications, these parameters are specified using command-line options. See the description of each individual application for more information.

Configuring Requester Programs (Applications)

All of the APPC Application Suite applications use LU 6.2 sessions to communicate with their partner applications on the destination nodes. To establish an LU 6.2 session with the partner application, you must supply the following parameters:

Remote TP name (RTPN)
> The *RTPN* enables the local application to identify the remote application. By default, each application uses the standard *RTPN* for its partner application, but you can override this value by using the application's command line options.

Mode name
> The local application must provide an SNA *mode name* for use by the session. The same *mode name* must be defined on the remote system. By default, each application provides a *mode name*, but you can override this value by using the application's command line options.

Partner LU name
> The local application must specify the LU on the remote system that handles requests for the remote TP. Specify the *partner LU name* (which is called the *RemoteSystemID* in the syntax statements) by specifying one of the following destination names on the command line:
> - *Fully qualified partner LU name* (*FQPLU*)
> - *CPI-C symbolic destination name*
> - **ANAME** *alias*

(For more information, see "Destination Names.")

Destination Names: Enter one of the following destination names on the command line to specify the *partner LU name* (which is called the *RemoteSystemID* in the syntax statements):

Fully qualified partner LU name (FQPLU)
> You can explicitly specify the *partner LU name* as the *FQPLU* in the form

NetworkName.LUName. In this case, the application either uses the default *mode name* and *RTPN* values or the values you specify by using the application's command line options.

For example, to establish an **APING** session with the LU called LUNAME7 on the SNA network NETID, enter the following on the command line:

aping NETID.LUNAME7

CPI-C symbolic destination name

You can supply the *partner LU name* by specifying a *CPI-C symbolic destination name*. The *CPI-C symbolic destination name* is the name of a CPI-C side information record; see *IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide* for details of how to configure this.

By using the *CPI-C symbolic destination name*, the application locates the appropriate side information record and uses the information contained in it to determine the partner LU address.

To establish an **APING** session with the partner LU specified in the side information record that has the symbolic destination name FRED, enter the following command:

aping FRED

ANAME *alias*

You can supply the *partner LU name* by specifying an **ANAME** *alias* as a *destination name*. This alias refers to a specific LU within the SNA network.

For example, enter the following command to associate the **ANAME** *alias* called max with the local control point LU:

aname reg -u max

Enter the following command to establish an **APING** session with the LU associated with the user called max:

aping max

For more information about **ANAME**, see "Using ANAME" on page 24.

Destination Names Search Order: The application searches for destination names in the following sequence:

1. *Fully qualified partner LU name (FQPLU)*

   If the *destination name* contains a period (**.**), the application interprets it as an *FQPLU*. If the application cannot locate the *FQPLU*, it does not continue to try to connect to any other LUs.

2. *CPI-C symbolic destination name*

   If the *destination name* does not contain a period (**.**), the application tries to locate a *CPI-C symbolic destination name* with that name.

3. **ANAME** *alias*

   If the application cannot locate a *CPI-C symbolic destination name* that matches the *destination name* you entered on the command line, it tries to obtain the **ANAME** *alias* from the **ANAME** provider program (**ANAMED**).

For more information about how the application obtains the alias from the **ANAME** record, see "Locating an ANAME Alias Using the ANAMED Provider Program" on page 29.

If these three methods fail, the application returns an error.

Configuring Provider Programs (Applications)

For a requester program to communicate with a provider program, Communications Server or one of its clients must automatically start the provider program (also known as an invokable TP) when required. Refer to the *IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide* for more information about configuring invokable TPs.

The **asuite.tps** file (stored in the directory **/etc/sna** (AIX) or **/etc/opt/ibm/sna** (Linux)) is included with the APPC Application Suite and provides the standard configuration for the provider programs (**AFTPD**, **ATELLD**, **AREXECD**, and **ANAMED**) on the server and on Remote API Clients on AIX or Linux. You can use this file directly with the **snatpinstall** program, or (on the server) you can configure the equivalent settings in the Motif administration program. For Remote API Clients on Windows, refer to the *IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide* in order to create an equivalent configuration file.

By default, the provider programs run with the user ID guest (AIX) or nobody (Linux). If you need to run the provider programs with user ID root, in order to use features that are available only to a System Administrator, the permissions of the provider programs must be changed to add the set-group-ID bit. For example to run **AREXECD** with root privilege, change the USERID entry in the **asuite.tps** file to root, and issue the following command:

chmod +s /usr/bin/arexecd (AIX)
chmod +s /opt/ibm/sna/bin/arexecd (Linux)

Note:

- The **ACOPY** program does not have its own provider program. Instead, it uses the **AFTPD** provider program.
- The **APING** program uses a provider program that is configured automatically as part of Communications Server.

Enter the following command to install **asuite.tps** on a server or on a Remote API Client on AIX or Linux:

snatpinstall -a asuite.tps

Sample asuite.tps File: The following is the standard **asuite.tps** file that is provided with APPC Application Suite.

For AIX:

```
[AFTPD]
PATH=/usr/bin/aftpd
ENV=APPCTPN=AFTPD
USERID=guest
GROUP=usr
TYPE=NON-QUEUED

[ATELLD]
PATH=/usr/bin/atelld
```

Configuring Your System for APPC Application Suite

```
ENV=APPCTPN=ATELLD
USERID=guest
GROUP=usr
TYPE=NON-QUEUED

[AREXECD]
PATH=/usr/bin/arexecd
ENV=APPCTPN=AREXECD
USERID=guest
GROUP=usr
TYPE=NON-QUEUED

[ANAMED]
PATH=/usr/bin/anamed
ENV=APPCTPN=ANAMED
USERID=guest
GROUP=usr
TYPE=NON-QUEUED
```

For Linux:

```
[AFTPD]
PATH=/opt/ibm/sna/bin/aftpd
ENV=APPCTPN=AFTPD
USERID=nobody
GROUP=nobody
TYPE=NON-QUEUED

[ATELLD]
PATH=/opt/ibm/sna/bin/atelld
ENV=APPCTPN=ATELLD
USERID=nobody
GROUP=nobody
TYPE=NON-QUEUED

[AREXECD]
PATH=/opt/ibm/sna/bin/arexecd
ENV=APPCTPN=AREXECD
USERID=nobody
GROUP=nobody
TYPE=NON-QUEUED

[ANAMED]
PATH=/opt/ibm/sna/bin/anamed
ENV=APPCTPN=ANAMED
USERID=nobody
GROUP=nobody
TYPE=NON-QUEUED
```

Chapter 2. AFTP

APPC File Transfer Protocol (**AFTP**) provides file manipulation and transfer services to application programs and end users. With **AFTP**, you can copy ASCII and binary files between your computer and any computer running the **AFTPD** provider program.

The **AFTP** application provides a common naming convention for files, a common directory structure, and a common set of file attributes. This common format enables you to locate and transfer files without knowing how the files are actually named and stored on the target computer. If the format of your files is supported by the operating system on each platform, you can share and store files among different platforms that support **AFTP**.

If either the source or the destination is an EBCDIC computer, the EBCDIC computer performs ASCII-to-EBCDIC translation when it receives a file from the ASCII computer, and EBCDIC-to-ASCII translation when it sends a file in the opposite direction.

In addition to transfer commands, the **AFTP** application has subcommands that help you locate the files with which you want to work. These subcommands enable you to:

* Set and query the current directory setting at the source computer and at the target computer
* List files in a selected directory at the source computer and at the target computer
* Set file transfer options

AFTP Configuration

You must configure the following files before using the **AFTP** application:

* **aftp.ini** (stored in the directory **/usr/lib/sna/samples** (AIX) or **/opt/ibm/sna/samples** (Linux))
* **asuite.tps** (stored in the directory **/etc/sna** (AIX) or **/etc/opt/ibm/sna** (Linux))

The **aftp.ini** initialization file is included with the **AFTP** and **ACOPY** applications. This file specifies masks for specific source and target files to ensure correct file renaming and proper transfer mode. On AIX / Linux systems, place this file in your home directory and rename it to **.aftp**.

The **aftp.ini** file maps a source mask for a particular type [for example, source_mask (***.ANNOUNCE**)] to a target mask [for example, target_mask (***.ANN**)]. In this example, a file on the source machine called **test.ANNOUNCE** is renamed to **test.ANN** on the target machine. This mapping ensures that correct file renaming is performed during the file transfer. The file type determines whether files are transferred in binary mode or the default ASCII mode. See "Sample aftp.ini Initialization File" on page 8, for a sample **aftp.ini** initialization file.

On Windows clients, the **aftp.ini** file is also used by the **AFTP** provider program to restrict access to files based on the conversation security user ID. See the sample **aftp.ini** file below for information about how to do this.

For information about **asuite.tps**, see "Configuring Provider Programs (Applications)" on page 5.

Sample aftp.ini Initialization File

A sample **aftp.ini** file is shown in the following figure:

```
/*
   This file must be copied to the user's $HOME directory and
   renamed to ".aftp"
*/
map source_mask(*.NONE)     target_mask(*.)        options();
map source_mask(*.LABEL)    target_mask(*.LBL)     options();
map source_mask(*.NONEBIN)  target_mask(*.)        options(binary);
map source_mask(*.ASSEMBLE) target_mask(*.A37)     options();
map source_mask(*.LISTPS)   target_mask(*.PS)      options(binary);
map source_mask(*.APLTF  )  target_mask(*.ATF)     options(binary);
map source_mask(*.EXEC   )  target_mask(*.EXC)     options();
map source_mask(*.PACKAGE ) target_mask(*.PKG)     options();
map source_mask(*.SCRIPT  )  target_mask(*.SCR)     options();
map source_mask(*.NEWS )    target_mask(*.NEW)     options();
map source_mask(*.DOCUMENT) target_mask(*.DOC)     options();
map source_mask(*.BOOK   )  target_mask(*.BOO)     options(binary);
map source_mask(*.ARC    )  target_mask(*.*  )     options(binary);
map source_mask(*.*BIN   )  target_mask(*.*[BIN])  options(binary);
map source_mask(*.BAS    )  target_mask(*.*  )     options(binary);
map source_mask(*.CFG    )  target_mask(*.*  )     options(binary);
map source_mask(*.COM    )  target_mask(*.*  )     options(binary);
map source_mask(*.EXE    )  target_mask(*.*  )     options(binary);
map source_mask(*.FLS    )  target_mask(*.*  )     options(binary);
map source_mask(*.LIB    )  target_mask(*.*  )     options(binary);
map source_mask(*.OBJ    )  target_mask(*.*  )     options(binary);
map source_mask(*.PIC    )  target_mask(*.*  )     options(binary);
map source_mask(*.RAM    )  target_mask(*.*  )     options(binary);
map source_mask(*.SYS    )  target_mask(*.*  )     options();
map source_mask(*.WKS    )  target_mask(*.*  )     options(binary);
map source_mask(*.WK1    )  target_mask(*.*  )     options(binary);
map source_mask(*.WK3    )  target_mask(*.*  )     options(binary);
map source_mask(*.ZIP    )  target_mask(*.*  )     options(binary);
map source_mask(*.AVAIL  )  target_mask(*.AVL)     options();
map source_mask(*.PLAS   )  target_mask(*.PLS)     options();
map source_mask(*.BMP    )  target_mask(*.*  )     options(binary);
map source_mask(*.DLG    )  target_mask(*.*  )     options(binary);
map source_mask(*.DLL    )  target_mask(*.*  )     options(binary);
map source_mask(*.ICO    )  target_mask(*.*  )     options(binary);
map source_mask(*.FNT    )  target_mask(*.*  )     options(binary);
map source_mask(*.PTR    )  target_mask(*.*  )     options(binary);
map source_mask(*.RES    )  target_mask(*.*  )     options(binary);
map source_mask(*.ANNOUNCE) target_mask(*.ANN)     options();
map source_mask(*.DOC    )  target_mask(*.DOC)     options();
map source_mask(*.CATALOG ) target_mask(*.CAT)     options();
map source_mask(*.IMG    )  target_mask(*.*  )     options(binary);
map source_mask(*.KPS    )  target_mask(*.*  )     options(binary);
map source_mask(*.KPL    )  target_mask(*.*  )     options(binary);
map source_mask(*.PAL    )  target_mask(*.*  )     options(binary);
map source_mask(*.PLX    )  target_mask(*.*  )     options();
map source_mask(*.R1P    )  target_mask(*.*  )     options(binary);
map source_mask(*.WRK    )  target_mask(*.*  )     options(binary);

/*
 * When running the AFTP provider program on a Windows client, the
 * provide_access keyword is used to provide access to certain
 * directories for named users.  Below are several sample provide_access
 * keywords.  All are commented out except the last one, which provides
 * access to the anonymous user to files under c:\tmp only.
 *
 */

/*
The following is an example of providing a user "johndoe" read/write
access to the files in d:\myfiles and its subdirectories.

provide_access
    users(johndoe)
    directory(d:\myfiles)
    permissions(read write)
    ;
*/

/*
The following examples show how to provide read-only access for
people who connect without security information ("anonymous").

provide_access
    users(anonymous)
    directory(c:\aftptest\)
    permissions(read)
```

Using AFTP

To enter the **AFTP** environment, ensure that the requester program (**AFTP**) is in a directory in your path, and type one of the following commands at the AIX / Linux prompt:

aftp

aftp *RemoteSystemID*

where *RemoteSystemID* is a Remote System ID as defined in "Destination Names" on page 3.

When **AFTP** is started, the **aftp>** prompt is displayed.

Alternatively, type the following at the AIX / Linux prompt to specify an input file that contains a series of **AFTP** subcommands:

aftp [**-f** *input-file*]

The commands in the input file are executed within the **AFTP** environment.

Before you can use **AFTP** subcommands to the **AFTPD** provider program, you first have to establish a session with that program. Some configurations also require you to specify security information for the provider program. For more information, see "Using AFTP (and ACOPY) When Security Is Required."

Selecting a User ID

File operations on the remote system are performed as a particular user. The default user is "anon". If the "anon" user ID has insufficient file access permissions on the remote machine or it is not configured, specify a different user ID by using the *UserID* parameter.

If conversation-level security is configured on the session, specify a user ID that is configured with conversation-level security on the remote system.

Using AFTP (and ACOPY) When Security Is Not Required

If conversation-level security is not required, the following conditions apply when using the **aftp** and **acopy** commands:

- If an "anon" account is available on a remote machine, you do not have to specify a *UserID* (see "Establishing a Session Using the Default User ID (ANON)" on page 11).
- If an "anon" account is not available, you must specify a *UserID*. In this case, a *Password* is required before you can perform file operations on the remote system (see "Establishing a Session Using a Specific User ID" on page 11).
- If you do not specify a *UserID* and an "anon" account is not available, you receive the following error message:

  The User ID is not available on this system

Using AFTP (and ACOPY) When Security Is Required

If conversation-level security is required, file operations on the remote system are permitted if the *UserID* you specify satisfies each of the following conditions:

- It is available on the remote system

- It is a user ID configured for conversation-level security

You must specify a *Password* before file operations are performed by the specified user account.

If the *UserID* you specify for the conversation-level security does not have a matching *UserID* on the remote system, file operations are done using the "anon" account on the remote computer. If the "anon" account does not exist, the copy fails.

Note: The System Administrator specifies the *UserID* and *Password* values by using the **define_userid_password** command. For AIX, if the System Administrator does not specify these values, they are taken from the user's **/etc/passwd** file.

Establishing a Session Using the Default User ID (ANON)

1. At the **aftp>** prompt, type:

   open *RemoteSystemID*

   where *RemoteSystemID* specifies the LU name of a computer on your APPC network that is running the **AFTPD** provider program. (For more information about specifying a destination, see "Destination Names" on page 3.)

2. At the user ID prompt, press **Return**.

   AFTP displays a message indicating that you are connected to *RemoteSystemID*.

Establishing a Session Using a Specific User ID

1. At the **aftp>** prompt, type:

   open *RemoteSystemID*

   where *RemoteSystemID* specifies the LU name of a computer on your APPC network that is running the **AFTPD** provider program. (For more information about specifying a destination, see "Destination Names" on page 3.)

2. At the user ID prompt, type a *UserID*.

3. At the password prompt, type the *Password* associated with the specified *UserID*.

   AFTP displays a message indicating that you are connected to *RemoteSystemID*.

Transferring Files

Using **AFTP**, you can transfer files from an **AFTP** target system to your local system, and you can transfer files from your local system to the target system. Some of the common transfer operations that **AFTP** supports are illustrated in the following examples:

Single file transfer without assigning a new file name

- To copy a remote file (in the remote current directory) to your current local directory without changing the file name, enter the following subcommand:

  get *RemoteFile*

- To copy a local file (in the local current directory) to the remote current directory without changing the file name, enter the following subcommand:

  put *Filename*

Single file transfer assigning a new file name

- To copy a remote file called *Filename*.doc (in the remote current directory) to the local current directory (and assign a new file name of *Filename*.txt), enter the following subcommand:

  get *Filename*.**doc** *Filename*.**txt**

- To copy a local file called *Filename*.txt (in the local current directory) to the remote current directory (and assign a new file name of *Filename*.doc), enter the following subcommand:

  put *Filename*.**txt** *Filename*.**doc**

Multiple file transfer using wildcards

- To copy multiple remote files (not in the remote current working directory) to a set of local files (not in the local working directory) and assign different file names, enter the following subcommand:

  get /work/a*.doc /u/a*.txt

  In this example, the **get** subcommand copies all documents starting with "a" and ending with a file extension of **.doc** in a remote directory called **/work/** to files with the extension **.txt** in a local directory named **/u/**.

- To copy multiple local files (not in the local current working directory) to a set of remote files (not in the remote working directory) and assign different file names, enter the following subcommand:

  put /u/a*.txt /work/a*.doc

  In this example, the **put** subcommand copies all documents starting with "a" and ending with a file extension of **.txt** in a local directory called **/u/** to files with the extension **.doc** in a remote directory named **/work/**.

If you want to be prompted for each file while transferring multiple files, issue the **prompt** subcommand before executing the transfer (see Table 3 on page 13).

See Table 4 on page 14, for a summary of file-transfer subcommands.

Ending a Session with a Remote System

To end the session with the current remote system while remaining in the **AFTP** environment, type **close**.

AFTP displays the message **Goodbye**, indicating that the session with the remote computer is closed. At this point, you can either establish a new session with a different partner or exit **AFTP**. (See "Exiting the AFTP Environment" on page 17.)

AFTP Subcommands

In addition to its file transfer capabilities, **AFTP** enables you to change the default settings for file transfers based on specific file transfer requirements, and to manipulate files and directories on the remote (target) system. The sections that follow summarize the uses of all of the **AFTP** subcommands. You can only execute these subcommands from within the **AFTP** command shell.

Opening and Closing a Connection to a Partner Computer

Table 2, summarizes the subcommands that enable you to open and close a connection to a partner computer.

Table 2. AFTP Subcommands to Start and End a Connection to a Remote Computer

AFTP Subcommand	Description
open *<destination>*	Opens a connection to *<destination>*, where *<destination>* is a Remote System ID as defined in "Destination Names" on page 3.
close	Closes the current open **AFTP** session without exiting from the **AFTP** environment.
disconnect or **disc**	Alias for **close.**

Setting or Querying AFTP File Transfer Attributes

You can define file transfer attributes from within the **AFTP** environment. When you start **AFTP**, the transfer attributes take default settings. You can change these settings based on the specific file transfer requirements. Once you set these attributes, they remain set until you update them or close your connection. An exception to this rule is the **modename** subcommand. The value of **modename** is not reset until you exit **AFTP**. Table 3, summarizes the subcommands that affect file transfer attributes.

Table 3. AFTP Subcommands to Set File Transfer Attributes

AFTP Subcommand	Description
ascii or **asc**	Changes the data transfer type to **ASCII**. Issue this subcommand only when a session is open.
bell [**on** │ **off**]	If your system supports this subcommand, issue **bell** to turn a bell (sounded during transmission) on or off. When you set **bell** to **on**, a sound is made after each file transfer completes. If you specify **bell** without a new setting, it toggles the current setting.
binary or **bin**	Changes the data transfer type to **binary**. Issue this subcommand only when a session is open.
date [**old** │ **new**]	Sets how the file date is handled during file transfers. If you set the **old** option (the default), all files you create inherit the date from the source file. If you set the **new** option, all files you create are time-stamped with the current date and time. If you specify **date** without an option, the current date setting is displayed. Issue this subcommand only when a session is open.
modename [*appc-modename*]	Sets the APPC mode name used in the transfer of files. The *appc-modename* parameter is one to eight characters long. The default APPC mode name is #BATCH. Issue this subcommand only when a session is closed.
prompt [**on** │ **off**]	Controls whether **AFTP** prompts for each file while executing a **get**, **put**, or **delete** subcommand. If you issue **prompt** without specifying an option, the current setting is displayed. The default setting is **off**.
status or **stat**	Provides information about the current transfer characteristics.
system or **sys**	Provides information about the server computer.

Table 3. AFTP Subcommands to Set File Transfer Attributes (continued)

AFTP Subcommand	Description
type [**ascii**	**binary**]	[**default**]	Sets the data transfer type to either **ASCII** or **binary**. If you specify **default** , **AFTP** checks the initialization file and uses the transfer type specified there. If no transfer type is specified in the initialization file, **AFTP** uses the **ASCII** transfer mode. The default setting for the **type** subcommand is **default**.

Transferring Files

The **AFTP** transfer function is described in "Using AFTP" on page 10.

Table 4, summarizes the **AFTP** subcommands that transfer files.

Table 4. AFTP Subcommands to Transfer Files

AFTP Subcommand	Description
get *remote-file* [*local-file*]	Use the **get** subcommand to transfer a copy of one or more files from the remote computer to the local computer. The *remote-file* parameter is the name of the file on the remote computer that you want to copy. You can use wildcards to create a list of files you want to copy. If you do not specify *local-file*, **AFTP** copies the files into your current local directory using the *remote-file* name without the directory path.
receive or **recv**	Alias for **get**.
put *local-file* [*remote-file*]	Use the **put** subcommand to transfer a copy of one or more files from the local computer to the remote computer. The *local-file* parameter is the name of the local file on your computer you want to copy to the remote computer. You can use wildcards to create a list of files you want to copy. If you do not specify *remote-file*, **AFTP** copies the files into the current directory on the remote computer using the *local-file* name without the directory path.
send	Alias for **put**.

Directory and File Manipulation on the Target Computer

AFTP includes subcommands to enable you to manipulate files and directories on the target computer, make and remove directories, list directory contents, rename files, and perform other common operating system functions.

Table 5, summarizes the **AFTP** subcommands used for directory and file manipulation on the target computer.

Table 5. AFTP Subcommands to Manipulate Directories and Files on the Target Computer

AFTP Subcommand	Description
cd *directory-spec*	Changes the current working directory of the target computer to the directory you specify with *directory-spec*.
delete *remote-file-spec* or **del** *remote-file-spec*	Deletes the file on the target computer you specify with *remote-file-spec*.

Table 5. AFTP Subcommands to Manipulate Directories and Files on the Target Computer (continued)

AFTP Subcommand	Description
dir [**file-spec**]	Provides a long directory list of the files and directories on the target computer. The list is displayed in the target computer's native format. Use **file-spec** to specify particular files you want to display. You can use wildcards to generate a list of files. If you specify a directory, all files within that directory on the target computer are displayed. Issue this subcommand only when a session is open.
ls [**file-spec**]	Displays a short directory list from the target computer. Only file names and directory names (not file attributes) are shown. Use **file-spec** to specify the files you want to display. You can use wildcards to generate a list of files. If you specify a directory, all files within that directory are displayed. Issue this subcommand only when a session is open.
lsd [**file-spec**]	Provides a short directory list from the target computer that contains only directories. Use **file-spec** to specify the directories you want displayed on the target computer. You can use wildcards to generate a list of files. If you specify a directory, all subdirectories within that directory are displayed. Issue this subcommand only when a session is open.
mkdir *directory-spec* or **md** *directory-spec*	Creates a new directory on the target computer. You cannot use wildcards. Issue this subcommand only when a session is open.
pwd	Displays the name of the present working directory on the target computer. Issue this subcommand only when a session is open.
rename *old-filename new-filename*	Renames a file on the target computer from *old-filename* to *new-filename*. You cannot use wildcards. Issue this subcommand only when a session is open.
rmdir *directory-spec* or **rd** *directory-spec*	Removes the directory on the target computer that you specify with *directory-spec*. You cannot use wildcards. Issue this subcommand only when a session is open.

Setting Allocation Sizes When Writing Files

AFTP includes subcommands that enable you to allocate sizes when writing files to, or reading files from, a file system that supports record-oriented files (such as VM, MVS™, and AS/400®). Use these subcommands to control block size, record length, and record format.

Table 6, summarizes the **AFTP** subcommands used for setting allocation sizes when writing files.

Table 6. AFTP Subcommands to Set Allocation Sizes when Writing Files

AFTP Subcommand	Description
alloc [*allocation-size*]	Sets the allocation size to use when writing files on a computer that uses record-oriented files, such as Virtual Machine (VM), Multiple Virtual Storage (MVS), and Application System 400 (AS/400) systems. The default setting is 0, indicating that the storing computer should use its own default setting.

Table 6. AFTP Subcommands to Set Allocation Sizes when Writing Files (continued)

AFTP Subcommand	Description
block [*block-size*]	Sets the block size to use when writing files. This value is used only on systems that use record-oriented files, such as VM, MVS, and AS/400 systems. The default setting is 0, indicating that the storing computer should use its own default setting.
lrecl [*record-length*]	Sets the record length to use when writing files. This value is used only on systems that use record-oriented files, such as VM, MVS, and AS/400 systems. The default setting is 0, indicating that the storing computer should use its own default setting.
recfm [*record-format*]	Sets the record format to use when writing files. This value is used only on systems that use record-oriented files, such as VM, MVS, and AS/400 systems. Following are the allowable values: 0, V, VA, VB, VM, VS, VBA, VBM, VBS, VSA, VSM, VBSA, VBSM, F, FA, FB, FM, FBA, FBM, FBS, FBSM, FBSA, U, and UA. The default setting is 0, indicating that the storing computer should use its own default setting.

Local Commands on the Source Computer

You can enter commands on the source computer at any time during an **AFTP** session.

You do not have to be connected to an **AFTPD** provider program to execute the subcommands summarized in Table 7.

Table 7, summarizes the subcommands that affect the source computer.

Table 7. AFTP Subcommands that Affect the Source Computer

AFTP Subcommand	Description
! *<command>*	Enables the user to issue miscellaneous non-**AFTP** commands on the local computer. Use *<command>* to specify any command you want to execute locally. Include any arguments in *<command>* that you want sent to the local command.
lcd [*pathspec*]	Changes the current directory on the local computer. If you do not specify *pathspec*, the command displays the current local directory path.
lpwd	Displays the current working directory of the local computer.

Using AFTP Help Subcommands

Table 8, summarizes the subcommands that enable you to access online help within the **AFTP** environment.

You do not have to be connected to an **AFTPD** provider program to execute the subcommands summarized in Table 8.

Table 8. AFTP Subcommands to Access Online Help

AFTP Subcommand	Description
?	Alias for **help**.

Table 8. AFTP Subcommands to Access Online Help (continued)

AFTP Subcommand	Description
help	Provides a list of topics for which help is available.
help *<command>*	Provides detailed help on the command you specify.

Exiting the AFTP Environment

Table 9, summarizes the subcommands that enable you to exit **AFTP**.

Table 9. AFTP Subcommands to Exit the AFTP Environment

AFTP Subcommand	Description
exit	Exits the **AFTP** environment, closing any active **AFTP** sessions.
bye	Alias for **exit**.
quit	Alias for **exit**.

AFTP Subcommands

Chapter 3. ACOPY

APPC Copy (**ACOPY**) is an application that enables you to send or receive a file between two computers (**ACOPY** does not support multiple file transfer). For a simple file transfer, **ACOPY** is quicker and easier to use than the transfer function included with **AFTP**.

ACOPY uses the following transaction programs:

- **ACOPY**, which runs on the source computer
- **AFTPD** (**ACOPY** uses the same provider executable file as **AFTP**; for more information about using the provider program, see "Using AFTP (and ACOPY) When Security Is Required" on page 10).

ACOPY Configuration

ACOPY uses **AFTPD** as its provider program, and it uses the same initialization and configuration files as **AFTP**. For more information about configuration, see "AFTP Configuration" on page 7.

Using ACOPY

Use the following command syntax to invoke **ACOPY**:

acopy [**Options**] *FromFile ToFile*

One file (either the *FromFile* or the *ToFile*) resides on the local machine, and the other file resides on the remote machine:

- The order in which you specify the files determines the direction of the copy (the first named file is copied to the second named file).
- Specify the local file as *filename*.
- Specify the remote file as *destination name:filename*. (For more information about specifying a *destination name*, see "Destination Names" on page 3.)

You use the same command to send and receive files. If the first named file is on the local node, **ACOPY** sends it to the remote node. If the first named file is on the remote node, **ACOPY** receives the file on the local node. Be sure that the parameter that identifies the remote file includes an identifier for the remote system.

For example, to send a local file named **name1.c** to a remote system named CPINET.LUNAME and save it as filename **name2.c**, enter the following:

acopy name1.c CPINET.LUNAME:
name2.c

In this example, CPINET.LUNAME (an identifier for the remote system) consists of an LUNAME on the CPINET network. The colon (**:**) following the LU name separates the remote system identifier from the file name. The identifier for the remote system can be a *fully qualified partner LU name* (as in the example), a *CPI-C symbolic destination name*, or an *alias* stored in the APPC Name Server (see "Destination Names" on page 3).

Enter the following command to receive the same file from the same remote system:

acopy CPINET.LUNAME:name2.c
name1.c

To limit access to **ACOPY** by unauthorized users, you can configure conversation-level and resource security for the copy of the provider program (**AFTPD**) that resides on your system (refer to the *IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide*). If security is configured for your system, you must specify the **-u** *UserID* option and the **-p** *Password* option. For information about these and other **ACOPY** options, see "ACOPY Options." See also "Using AFTP (and ACOPY) When Security Is Required" on page 10.

ACOPY Restrictions

Following are the restrictions for using **ACOPY**:

- You can specify only one file to copy from and one file to copy to. To perform more complex copying operations, use **AFTP**.
- One of the two named files must be a remote file, and the other must be a local file. It is not possible to copy files local-to-local or remote-to-remote.
- To use the system without specifying a user ID and password, you must set up an "anon" account and disable conversation-level security. (For more information about disabling security, refer to the *IBM Communications Server for Data Center Deployment on AIX or Linux Administration Guide*.)

ACOPY Options

The **ACOPY** options are as follows:

-u *UserID*

Specifies the user ID that is sent to the remote system. The *UserID* can be one to eight characters in length. One indication that the remote transaction program requires security is if you receive the CPI-C error message **CM_SECURITY_NOT_VALID** when attempting to copy a file. If you have conversation-level security configured for your system, this option is required.

-p *Password*

Specifies the password that is sent to the remote system. This option is required when a user ID has been specified or if you have security configured for your system. The *Password* can be one to eight characters in length. If you do not enter a password after specifying a user ID, **ACOPY** will prompt you to specify one.

-m *ModeName*

The transfer mode name. The default *ModeName* is #BATCH.

-t *TPName*

The TP name that runs on the remote system (provider TP). The default *TPName* is **AFTPD**.

-n

This option overrides the security option. By default, **ACOPY** sends the user ID of the running process to the remote system. If the remote system is not configured to require security, this option stops a user ID from being sent. If the remote system is not configured to require security, use this option if you receive the error message **CM_SECURITY_NOT_VALID**.

-b   This option specifies that data should be transferred in binary format.

-a   This option specifies that data should be transferred in ASCII text format. This is the default file transfer mode, but you can configure the transfer mode default to be either ASCII or binary depending on the file type being transferred. For more information, see "AFTP Configuration" on page 7.

If you specify any of the options more than once, the system uses the last one entered.

Getting Help

To get help on using **ACOPY**, enter either of the following commands:

acopy ?

acopy -h

acopy -help

Using ACOPY

Chapter 4. ANAME

The APPC Name (**ANAME**) application is a program that simplifies naming in an SNA network.

Although it is necessary to use LU names to identify a partner in a transaction program, it can be difficult to associate LU names with the names of the people or resources using those LUs. **ANAME** eliminates the user's need to know LU names by mapping the names to more usable nicknames (user names) that are accessible across the network. For example, rather than using a fully qualified LU name, such as ABCDEFG.NR10691, you can map it to a simpler user name such as PrintServer.

You can also use the group membership function of **ANAME** to categorize LUs according to common function or group association. Users can also be associated with the transaction programs they use.

The **ANAME** program provides the following functions, which are referred to collectively as "Name Services:"

- Adding a name to the **ANAME** database
- Removing a name from the **ANAME** database
- Viewing records in the **ANAME** database

An **ANAME** network includes one provider program **ANAMED** that maintains the database and provides the services, and one or many requester programs that request the services.

ANAME Configuration

To configure **ANAME**, do the following:

- Modify the **asuite.tps** file (stored in the directory **/etc/sna** (AIX) or **/etc/opt/ibm/sna** (Linux)), which provides the standard configuration for the provider TPs. For more information, see "Configuring Provider Programs (Applications)" on page 5 and "Sample asuite.tps File" on page 5.
- Modify the sample initialization file called **anamed.ini** (stored in the directory **/usr/lib/sna/samples** (AIX) or **/opt/ibm/sna/samples** (Linux)). Rename this file to **.anamed** and store it in the home directory of the user that is running the **ANAMED** program (this user is configured in the **asuite.tps** file).
- Create a side information record (see "Ensuring Name Resolution" on page 26).

Modifying the anamed.ini Initialization File

The **anamed.ini** initialization file is included with the **ANAME** application. Before using **ANAME**, do the following:

- Provide values for the *set_system_administrator* LU name, (which refers to the FQLU name on the user's network used by the System Administrator), and *set_database_path*. (The **set_database_path** parameter sets the directory in which the **ANAME** database is stored. The default directory is the home directory of the **ANAME** user.)
- Rename the file to **.anamed** and copy it to the home directory of the user configured to run the provider program.

Sample anamed.ini Initialization File

The initialization file is a text file that identifies the System Administrator and the database path. Modify the sample **anamed.ini** initialization file as described in "Modifying the anamed.ini Initialization File" on page 23. The standard file is shown in Figure 2.

```
/*
 * This is a sample ANAMED.INI file.
 *
 * The following keys should be changed when configuring the
 * NameServer target system.
 *
 * This file should be placed in the target's $HOME directory and renamed
 * to ".anamed"
 *
 * See the NameServer system administrator's guide for more
 * information.
 */

/*
 * Change the set_system_administrator value to the fully-qualified
 * LU name of the NameServer administrator's computer.
 */
  set_system_administrator
      lu_name( SysAdmin.LU );
/*
 * Change the set_database_path value to the drive and path of the
 * NameServer database file.
 */
  set_database_path
      directory( . );
```

Figure 2. Sample anamed.ini File

Using ANAME

The **ANAME** database contains all of the names that are available on the **ANAME** network.

- The location of the database is identified in the **anamed.ini** file.
- Users can create and delete entries associated with their local LU.
- The administrator for **ANAME** can manage entries for any user.

The four fields that are contained in a record in the database are as follows:

User name
Specify this value when you register with **ANAME**. Your user name should be an easy-to-remember, unique nickname for the fully qualified LU name that identifies you to others in the **ANAME** network.

Fully qualified LU name
The fully qualified LU name associated with this particular record. Communications Server takes this from the name of the local LU that you are using with the **ANAME** application. This is normally the default LU associated with the local node (the control point LU), unless you have set the APPCLLU environment variable to specify a particular local LU. Alternatively, if you are logged in as the System Administrator, you can use the **-f** *LUname* option on the command line to specify a particular local LU name.

Group name
Optionally, specify this value when you register with **ANAME**. There are no restrictions on the number of instances of the same group name or on

the number of users registered as members of one group. Several users can register with the same group name. Typically, all members of a department or team register as part of a particular group.

Transaction program (TP) name

Optionally, specify this value when you register with **ANAME**. This field identifies the TPs associated with a fully qualified LU name. There are no restrictions on the number of instances of the same TP name or the number of users registered with one TP name.

Note: Commands for **ANAME** functions can be entered in either uppercase or lowercase. For example, to register a user name, you can use either the **REG** or **reg** command.

Names registered with **ANAME** should be values that are easy to remember and will not change over time. When selecting names to register with **ANAME**, consider how the names will be used as the **ANAME** network grows and changes. Follow these rules and conventions:

- Use names that will remain unique within a specific domain. For example, avoid using first names and initials.
- Use defined and understood domains as group names, such as team or department names.
- Use names that are easily associated with the person or entity using the name, such as local area networks (LANs) or host IDs.
- Choose names that can be used for a variety of purposes throughout APPC. For example, because APPC partner aliases must be eight characters or less in length, an **ANAME** user name that is also used as an alias should meet this size limitation.

Adding a Record

It is not necessary to enter information in all four fields when adding a record. What you enter depends on your situation.

- User name

  The simplest way to register a user name in the **ANAME** database is to associate a name you select with a fully qualified LU name. To do so, enter the following:

  aname REG -u *UserName*

  A record is created in the database associating the *UserName* with the *fully qualified LU name* from which the command was issued. The *UserName* must be unique to the *fully qualified LU name*.

- Group name or names

  If you belong to any group or groups in the **ANAME** network, add the group names to the database so that others on the network can find you. (You can use the same user name if you are using the same *fully qualified LU name*.) For example, if you belong to a team and to a department, you can register both by entering the following:

  aname REG -u *UserName* **-g** *TeamName*

  aname REG -u *UserName* **-g** *DepartmentName*

- Transaction program names

If you want other users on the **ANAME** network to know what transaction programs are installed on your computer, you can register the program names in the database. For example, to register the **ATELLD** program in the database and associate it with your *fully qualified LU name*, enter the following command:

aname REG -t ATELLD

Because it is possible to have more than one user name for each *fully qualified LU name*, you can make the association more specific by adding a user name to the command:

aname REG -u *UserName* **-t ATELLD**

For more information, see "Summary of Supported ANAME Commands" on page 29.

Ensuring Name Resolution

The **ANAME** application must be able to locate the **ANAMED** provider program to access the name-to-PLU mapping (to locate the PLU). You can accomplish this in one of two ways:

- Specify the **-s** option on the command line when adding a record (see Table 12 on page 29) to locate the provider program (**ANAMED**).
- Create a side information record to locate the provider program (**ANAMED**).

  For example, if the **ANAMED** provider program was on the LU called MYLU on the network called MYNETWORK, the following creates an appropriate side information record:

  snaadmin define_cpic_side_info, sym_dest_name=ANAMED, partner_lu_name=MYNETWORK.MYLU, tp_name=ANAMED

Character Sets

Valid characters for all fields in **ANAME** include uppercase and lowercase A–Z, the numbers 0–9, and the special characters listed in Table 10.

Table 10. Additional Valid Characters for All Fields in ANAME

	Blank	,	Comma
"	Double quote	-	Hyphen
#	Number sign	/	Front slash
$	Dollar sign	:	Colon
%	Percent sign	;	Semicolon
&	Ampersand	<	Less than sign
'	Single quote	=	Equal sign
(	Left Parentheses	>	Greater than sign
)	Right parentheses	@	At sign
+	Plus sign	_	Underscore

Several non-alphanumeric characters have restrictions, as follows:

- The hyphen (-) cannot be used as the first character of any field value.
- The asterisk (*) and question mark (?) characters cannot be used in field values. They are reserved as special characters for search routines. (See "Wildcards" on page 28.)
- The period (.) character cannot be used as a value in group names or user names.

Removing a Name from the ANAME Database

The `delete` function enables you to remove a record or records from the **ANAME** database. Records are removed only if the *fully qualified LU name* in the record matches the LU that you are currently using with the requester program. For example, to delete all the records with the specified user name and the requester program's current *fully qualified LU name*, enter:

aname DEL -u *UserName*

To delete a record with a specified user name and group name, enter:

aname DEL -u *UserName* **-g** *GroupName*

For more information, see "Summary of Supported ANAME Commands" on page 29.

Viewing Names in the ANAME Database

You can use the `query` function to view a single record or multiple records in the **ANAME** database.

For example, to view the record of a group called TechDept, enter:

aname QRY -g TechDept

To view all the records for a particular user name, enter:

aname QRY -u *UserName*

To view all the records that match a specific user name and group name, enter:

aname QRY -u *UserName* **-g** *GroupName*

To view all the records that match a specific group name and TP name, enter:

aname QRY -g *GroupName* **-t** *TPName*

Note:

- If you specify only one field, the program returns entries that match on this field and have no other fields set, in preference to entries that match on this field and also have other fields set. For example, if you specify only the user name, and there are records for the specified user name both with and without a group name specified, this command returns only the records that have no group name specified.
- If you specify a TP name, the program returns entries that contain this TP name, and also entries that have no TP name specified.
- When searching the database for a specific value, only TP names are case sensitive. Matches for the other three fields are found by alphabetic value only.

To view multiple records that contain a common field value, use wildcard searches. (For more information about wildcards, see "Wildcards" on page 28.) For example, to view all the records for user names that begin with the letter "A", enter:

aname QRY -u "A*"

For more information, see "Summary of Supported ANAME Commands" on page 29.

Using the Help Option

To obtain help on how to use all **ANAME** commands, enter one of the following commands:

aname ?

aname -h

aname -help

To obtain help on a specific command, include the function name in the command. For example, to get help for the Register function, enter one of the following commands:

aname REG ?

aname REG -h

aname REG -help

For more information, see "Summary of Supported ANAME Commands" on page 29.

Duplicating a User Name

A System Administrator can duplicate a *UserName* that is already registered by entering the following command:

aname reg -d *-u UserName*

This command can be used to associate an existing user name with resources on local systems and those resources on remote systems. This enables a remote user to request any of the resources on the local node by using a single user name. The **ANAMED** provider program then determines which LU handles requests for the resource.

Wildcards

Because some AIX / Linux shells do wildcard expansion on the command line, it is important to remember to put quotation marks (") around wildcard commands. The following set of wildcards is recognized by **ANAME**:

Full wildcard
> Use the asterisk (∗) to view or delete all records in a particular field. For example, **-u** "∗" indicates all user names.

Partial wildcard
> Use the asterisk (∗) following a value to indicate that you want to view or delete all the values that contain the characters you specify. For example, **-u**"J∗" indicates all the values that begin with the letter "J".

Placeholder wildcard
> Use the question mark (?) to indicate a placeholder. A question mark between characters causes a search on the specified characters and any

unspecified character that falls in between. For example, **-u**"J??n" would match values such as "John", "Jean", and "Joan".

Summary of Supported ANAME Commands

Table 11, summarizes the commands that enable you to control the **ANAME** database. Use each command with at least one of the flags shown in Table 12.

You can enter the following commands in uppercase or lowercase:

Table 11. Supported ANAME Commands

ANAME Command	Description
register or **reg** or **r**	Registers a name in the **ANAME** database.
delete or **del** or **d**	Removes a name from the **ANAME** database.
query or **qry** or **q**	Enables you to view records in the **ANAME** database. You may use wildcards to broaden the query criteria.

Table 12, summarizes the options and parameters that you can use with the **ANAME** commands.

Table 12. Options and Parameters Used with ANAME Commands

ANAME Option/Parameter	Description
-u *username*	Specifies a user name.
-g *groupname*	Specifies a group name.
-t *tpname*	Specifies a TP name.
-f *LUname*	Specifies an LU name. Only the System Administrator (as defined in the **.anamed** initialization file) has authorization to use this option.
-d	Duplicates a user record with a different LU name. Only the System Administrator (as defined in the **.anamed** initialization file) has authorization to use this option.
-h, **-help**, or **-?**	Provides help for the command.
-s *RemoteSystemID*	Identifies the LU on which **ANAMED** is running. By default, the *RemoteSystemID* is the LU identified in the CPI-C side information record stored on each machine. The *RemoteSystemID* can be either a *CPI-C symbolic destination name* or a *fully qualified LU name*.
-a *UserID*	Specifies the user ID to be sent to the partner computer. This option should be used if conversation-level security is required for the link.
-p *password*	Specifies the user's password. This option is required if the **-a** option is used. Only use this option if conversation-level security is active on the link.

Locating an ANAME Alias Using the ANAMED Provider Program

When you specify a valid **ANAME** *alias* as the *destination name* on the command line, the application (**AFTP**, **ACOPY**, **APING**, **AREXEC**, or **ATELL**) tries to connect to the provider program for **ANAME** (**ANAMED**). To establish a successful connection, you must have already created a side information record and specified ANAMED as the *symbolic destination name*. The side information record contains the *FQPLU* name on which the **ANAMED** provider program is running.

Locating an ANAME Alias Using the ANAMED Provider Program

If you did not create this side information record, the application is unable to locate the **ANAMED** provider program and, therefore, is unable to locate the partner LU. If the application is able to locate the **ANAMED** provider program but the **ANAME** *alias* is not associated with an LU, the application is also unable to locate the partner LU.

Chapter 5. APING

APPC Ping (**APING**) is a program that exchanges data packets with a partner computer and measures how long the data transfer takes. **APING** is often the first program configured and run after configuring Communications Server because it is a simple way to check connectivity between your local Communications Server computer and remote computers.

APING includes one transaction program—**APING**, the requester program. (There is no separate executable program for the partner program for **APING** because it is built into Communications Server.)

Note: It is possible to disable the **APING** partner program that is built into Communications Server by using the **snaadmin define_node** command. To do this, ensure that the node is inactive, then set the *ptf_flags* parameter to external_apingd. (For more information about the **snaadmin define_node** command, refer to the *IBM Communications Server for Data Center Deployment on AIX or Linux Administration Command Reference*.)

Use **APING** to determine whether you can establish a session between two computers. If session allocation fails, **APING** displays extensive error information. You can also use **APING** to get an approximate measure of the session setup time between two computers, and of the throughput and turnaround time on that APPC session.

APING Configuration

No configuration is required for the **APING** application.

Using APING

Use the following command syntax to invoke **APING**:

aping Options *RemoteSystemID*

The *RemoteSystemID* can be either a *CPI-C symbolic destination name*, *a fully qualified LU name*, or an **ANAME** *alias*. If an **ANAME** *alias* conflicts with a *CPI-C symbolic destination name*, the APPC Application Suite uses the *CPI-C symbolic destination name*. For more information, see "Destination Names" on page 3.

The simplest way to use **APING** is to specify only the *RemoteSystemID*. For example, you can start **APING** with the following command:

aping *RemoteSystemID*

Running the **APING** program results in a session allocation that is timed. When the session allocation is complete, **APING** sends 100 bytes to the partner computer and receives the same number of bytes. **APING** then displays how long it took to complete the transfer. Because the first timing usually includes process start up time on the partner side, the data exchange occurs twice.

The following example shows some additional parameters that can be used with
APING (for more information about these parameters, see "APING Parameters
and Options"):

aping -s 10000 -i 2 -c 4 *RemoteSystemID*

In this example:
   -s 10000 specifies the size of the packet transmitted.
   -i 2 requests two iterations.
   -c 4 specifies the number of Send calls by each side before giving the partner
   permission to send.

The next figure, Figure 3, shows sample output from the **APING** program.

```
Allocate duration:                                     0 ms
Program startup and Confirm duration:               720 ms

        Duration          Data Sent         Data Rate          Data Rate
         (msec)            (bytes)            (KB/s)             (mb/s)
        --------          ---------         ---------          ---------
            60              64000            1041.7              8.333
            30              64000            2083.3             16.667
Totals:     90             128000            1388.9             11.111

Duration statistics:  Min = 30        Ave = 45         Max =60
```

Figure 3. Sample Output for APING

The value shown for the *Allocate duration* field in Figure 3, is the length of time it
takes the Allocate call to return to the program. The next call in **APING** is a
Confirm call, which is timed to determine the approximate program start up time
of the partner transaction program.

The table output in Figure 3, shows the duration and data transfer statistics for
each "ping" (Send and Receive cycle):

• The number of ping duration lines corresponds to the number of iterations
  requested by using the *-i* parameter. (See "APING Parameters and Options.")
  This example uses two iterations, which is the default.

• *Totals* shows the overall data rate for all pings.

• *Duration statistics* shows a summary of the ping duration lines already displayed.
  This line shows the minimum, average, and maximum ping duration in
  milliseconds.

APING Parameters and Options

You can use the following parameters and options with the **aping** command:

RemoteSystemID
        This parameter identifies the partner computer on which the **APINGD**
        provider program is running. The *RemoteSystemID* (*partner LU name*) can
        be a *fully qualified partner LU name* (*FQPLU*), a *CPI-C symbolic destination
        name*, or an **ANAME** *alias*.

        For more information about *RemoteSystemID*, see "Destination Names" on
        page 3.

-m *ModeName*
        The mode name. The default mode name is #INTER.

-s *Size*   This option specifies the size, in bytes, of the packet to be transmitted. The default is 100 bytes.

-i *N*   *N* is the number of iterations made. The default number of iterations is 2. The number of iterations is shown in the output as the number of sets of timing information. You may want to increase this number to get a larger sample of timings. Allowable values are 1 to 32767.

-c *N*   *N* is the number of consecutive packets sent by each side before giving the partner permission to send. For each iteration, each side sends this number of packets, with each one being of the specified packet size. Allowable values are 1 to 32767. The default number of consecutive packets is 1.

-u *UserID*

The user ID that is sent to the partner computer. The user ID can be up to eight characters in length. Use this parameter if the *RemoteSystemID*transaction program is configured for security.

-P *Password*

This is the password for the user ID that is sent to the partner computer. The password, which can be one to eight characters in length, is required when a user ID has been specified. If you do not enter a password after specifying a user ID, **APING** will prompt you to enter one.

-r   Randomize the outgoing data buffer. By default, the outgoing data is all zeros.

-#   Output a count of completed transactions along with the timing information.

-q   Quiet operation. No output is produced. If the command succeeds, a return code of zero is returned. If the command fails, a nonzero return code is returned, and error information is written to the **APING** error log (**aping.err**).

-n   Overrides the security parameter. By default, a request for the remote application should include a user ID and an indication that the password has been verified. If the destination accepts already-verified user IDs, the user ID of the running process is sent without a password. If the destination (provider) transaction program is not configured to require security, use the **-n** option if you receive the error message **XC_SECURITY_NOT_VALID** .

-1   This option specifies that data be sent only from the requester TP to the provider TP (not in both directions).

Note: This option is the numeral one (1), not the letter "L".

If you specify any of the options more than once, the system uses the last one entered.

Using Help

To get help on how to use **APING**, enter one of the following commands:

aping ?

aping -h

aping -help

Using APING

Chapter 6. AREXEC

APPC Remote EXECution (**AREXEC**) enables you to execute any single command on a remote computer. All output of the command that goes to standard output or standard error output is also routed to your screen.

AREXEC uses the following transaction programs:

- **AREXEC**, which runs on the local computer (requester TP)
- **AREXECD**, which runs on the remote computer (provider TP)

AREXEC Configuration

If you have not already done so, install the **asuite.tps** file as described in "Configuring Provider Programs (Applications)" on page 5. This file provides the standard configuration for the provider TPs.

Using AREXEC

Use the following command syntax to invoke **AREXEC**:

arexec [*Options*] *RemoteSystemID Command*

The simplest way to use **AREXEC** is to specify only the *RemoteSystemID* (see "Destination Names" on page 3) and the command. For example, you can start **AREXEC** in the following way:

arexec *RemoteSystemID* ls *.c

This command sends the command **ls *.c** to the specified remote computer. **AREXECD**, the **AREXEC** partner program on the remote computer, executes the command, and displays the results as the output of **AREXEC** on the local computer. If you specify any of the options more than once, the system uses the last one entered.

Security parameters may also be required, depending on how your system is configured.

AREXEC Parameters and Options

The **AREXEC** parameters and options are as follows:

RemoteSystemID
> This parameter identifies the partner computer on which the **AREXECD** provider program runs. The *RemoteSystemID* (*partner LU name*) can be a *fully qualified partner LU name* (*FQPLU*), a *CPI-C symbolic destination name*, or an **ANAME** *alias*.
>
> For more information about *RemoteSystemID*, see "Destination Names" on page 3.

Command
> The command string to be executed on the remote computer. For example, if the user ID is anon, then a command of **ls** would list the contents of anon's home directory, **/home/anon**.

-m *ModeName*

The mode name. The default mode name is `#INTER`.

-t *TPName*

The name of the partner TP. The default TP name is `AREXECD`.

-u *UserID*

The user ID that is sent to the partner computer. The user ID can be up to eight characters in length. Use this option if the remote transaction program is configured for security.

-p *Password*

This is the password for the user ID that is sent to the partner computer. The password, which can be one to eight characters in length, is required when a user ID has been specified. If you do not enter a password after specifying a user ID, **AREXEC** will prompt you to do so.

-n

This option overrides the security option. By default, a request for the remote application should include a user ID and an indication that the password has been verified. If the destination accepts already-verified user IDs, the user ID of the running process is sent without a password. If the remote transaction program is not configured to require security, use this option if you receive the error message **XC_SECURITY_NOT_VALID** .

-q

Quiet operation. The **AREXEC** banner is not displayed; only the results from the command are shown.

Getting Help

To obtain help on using **AREXEC**, enter one of the following commands:

arexec ?

arexec -h

arexec -help

Chapter 7. ATELL

APPC Tell (**ATELL**) enables a workstation user to send a message to a workstation user on another machine.

ATELL consists of two transaction programs:
- **ATELL**, which runs on the local machine (requester TP)
- **ATELLD**, which runs on the remote machine (provider TP)

ATELL Configuration

If you have not already done so, install the **asuite.tps** file as described in "Configuring Provider Programs (Applications)" on page 5. This file provides the standard configuration for the provider TPs.

Using ATELL

Use the following command syntax to invoke **ATELL**:

atell [**Options**] [*UserID@*]*RemoteSystemID Message*

The simplest way to use **ATELL** is to specify only the destination and the message. For example:

atell *RemoteSystemID* **Good Morning**

This command sends the message Good Morning to the specified destination computer, where it is displayed. The method used for displaying the message may vary depending on the remote operating system.

Security parameters may be required, depending on the configuration of your system.

ATELL Parameters and Options

Following are the **ATELL** parameters and options:

RemoteSystemID
> This parameter identifies the remote computer to which you want to send your message and where the **AREXECD** provider program runs.
>
> The *RemoteSystemID* (*partner LU name*) can be a *fully qualified partner LU name (FQPLU)*, a *CPI-C symbolic destination name*, or an **ANAME** *alias*. If the *RemoteSystemID* is a *CPI-C symbolic destination name*, it can be one to eight characters in length and must be configured on the local system.
>
> For more information about *RemoteSystemID*, see "Destination Names" on page 3.
>
> To send a message to a particular user on a system, use the following format:
>
> *UserID@RemoteSystemID*

Message

The message text to be sent to the remote computer.

-m *ModeName*

The mode name. The default mode name is `#INTER`.

-t *TPName*

The TP name that runs on the remote machine. The default TP name is `ATELLD`.

-u *UserID*

The user ID that is sent to the remote computer. The user ID can be one to eight characters in length. Use this option if the remote transaction program is configured for security.

-p *Password*

The password for the user ID that is sent to the remote computer. This option is required when the user ID is specified. If you do not enter a password after specifying a user ID, **ATELL** prompts you to do so. The password can be one to eight characters in length.

-n

This option overrides the security parameter. By default, a request for the remote application should include a user ID and an indication that the password has been verified. If the destination accepts already-verified user IDs, the user ID of the running process is sent without a password. If the remote transaction program is not configured to require security, use this option if you receive the error message **XC_SECURITY_NOT_VALID** .

If you specify any of the options more than once, the system uses the last one entered.

Getting Help

To get help on using **ATELL**, enter one of the following commands:

atell ?

atell -h

atell -help

Appendix. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
>   IBM Director of Licensing
>   IBM Corporation
>   North Castle Drive
>   Armonk, NY 10504-1785
>   U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:
>   Intellectual Property Licensing
>   Legal and Intellectual Property Law
>   IBM Japan, Ltd.
>   19-21, Nihonbashi-Hakozakicho, Chuo-ku
>   Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR 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.

This information 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 without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

    Site Counsel
    IBM Corporation
    P.O. Box 12195
    3039 Cornwallis Road
    Research Triangle Park, North Carolina 27709-2195
    U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE: This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:

® (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. ® Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at Copyright and trademark information at www.ibm.com/legal/copytrade.shtml.

Intel is a registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Adobe and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Bibliography

The following IBM publications provide information about the topics discussed in this library. The publications are divided into the following broad topic areas:

- IBM Communications Server for AIX
- IBM Communications Server for Linux
- Systems Network Architecture (SNA)
- Advanced Program-to-Program Communication (APPC)
- Programming

For IBM Communications Server for AIX and IBM Communications Server for Linux books, brief descriptions are provided. For other books, only the titles and order numbers are shown here.

IBM Communications Server for AIX Publications

The IBM Communications Server for AIX library comprises the following books. In addition, softcopy versions of these documents are provided on the CD-ROM. See *IBM Communications Server for AIX Quick Beginnings* for information about accessing the softcopy files on the CD-ROM. To install these softcopy books on your system, you require 9–15 MB of hard disk space (depending on which national language versions you install).

- *IBM Communications Server for AIX Migration Guide* (SC31-8585)

  This book explains how to migrate from Communications Server for AIX Version 4 Release 2 or earlier to IBM Communications Server for AIX Version 6.

- *IBM Communications Server for AIX Quick Beginnings* (GC31-8583)

  This book is a general introduction to IBM Communications Server for AIX, including information about supported network characteristics, installation, configuration, and operation.

- *IBM Communications Server for AIX Administration Guide* (SC31-8586)

  This book provides an overview of SNA and IBM Communications Server for AIX, and information about IBM Communications Server for AIX configuration and operation.

- *IBM Communications Server for AIX Administration Command Reference* (SC31-8587)

  This book provides information about SNA and IBM Communications Server for AIX commands.

- *IBM Communications Server for AIX or Linux CPI-C Programmer's Guide* (SC23-8591)

  This book provides information for experienced "C" or Java programmers about writing SNA transaction programs using the IBM Communications Server CPI Communications API.

- *IBM Communications Server for AIX or Linux APPC Programmer's Guide* (SC23-8592)

  This book contains the information you need to write application programs using Advanced Program-to-Program Communication (APPC).

- *IBM Communications Server for AIX or Linux LUA Programmer's Guide* (SC23-8590)

  This book contains the information you need to write applications using the Conventional LU Application Programming Interface (LUA).

- *IBM Communications Server for AIX or Linux CSV Programmer's Guide* (SC23-8589)

  This book contains the information you need to write application programs using the Common Service Verbs (CSV) application program interface (API).

- *IBM Communications Server for AIX or Linux MS Programmer's Guide* (SC23-8596)

  This book contains the information you need to write applications using the Management Services (MS) API.

- *IBM Communications Server for AIX NOF Programmer's Guide* (SC31-8595)

  This book contains the information you need to write applications using the Node Operator Facility (NOF) API.

- *IBM Communications Server for AIX Diagnostics Guide* (SC31-8588)

  This book provides information about SNA network problem resolution.

- *IBM Communications Server for AIX or Linux APPC Application Suite User's Guide* (SC23-8595)

  This book provides information about APPC applications used with IBM Communications Server for AIX.

- *IBM Communications Server for AIX Glossary* (GC31-8589)

  This book provides a comprehensive list of terms and definitions used throughout the IBM Communications Server for AIX library.

IBM Communications Server for Linux Publications

The IBM Communications Server for Linux library comprises the following books. In addition, softcopy versions of these documents are provided on the CD-ROM. See *IBM Communications Server for Linux Quick Beginnings* for information about accessing the softcopy files on the CD-ROM. To install these softcopy books on your system, you require 9–15 MB of hard disk space (depending on which national language versions you install).

- *IBM Communications Server for Linux Quick Beginnings* (GC31-6768 and GC31-6769)

  This book is a general introduction to IBM Communications Server for Linux, including information about supported network characteristics, installation, configuration, and operation. There are two versions of this book:
      GC31-6768 is for IBM Communications Server for Linux on the i686, x86_64, and ppc64 platforms
      GC31-6769 is for IBM Communications Server for Linux for System z.

- *IBM Communications Server for Linux Administration Guide* (SC31-6771)

  This book provides an overview of SNA and IBM Communications Server for Linux, and information about IBM Communications Server for Linux configuration and operation.

- *IBM Communications Server for Linux Administration Command Reference* (SC31-6770)

  This book provides information about SNA and IBM Communications Server for Linux commands.

- *IBM Communications Server for AIX or Linux CPI-C Programmer's Guide* (SC23-8591)

  This book provides information for experienced "C" or Java programmers about writing SNA transaction programs using the IBM Communications Server CPI Communications API.

- *IBM Communications Server for AIX or Linux APPC Programmer's Guide* (SC23-8592)

This book contains the information you need to write application programs using Advanced Program-to-Program Communication (APPC).
- *IBM Communications Server for AIX or Linux LUA Programmer's Guide* (SC23-8590)

  This book contains the information you need to write applications using the Conventional LU Application Programming Interface (LUA).
- *IBM Communications Server for AIX or Linux CSV Programmer's Guide* (SC23-8589)

  This book contains the information you need to write application programs using the Common Service Verbs (CSV) application program interface (API).
- *IBM Communications Server for AIX or Linux MS Programmer's Guide* (SC23-8596)

  This book contains the information you need to write applications using the Management Services (MS) API.
- *IBM Communications Server for Linux NOF Programmer's Guide* (SC31-6778)

  This book contains the information you need to write applications using the Node Operator Facility (NOF) API.
- *IBM Communications Server for Linux Diagnostics Guide* (SC31-6779)

  This book provides information about SNA network problem resolution.
- *IBM Communications Server for AIX or Linux APPC Application Suite User's Guide* (SC23-8595)

  This book provides information about APPC applications used with IBM Communications Server for Linux.
- *IBM Communications Server for Linux Glossary* (GC31-6780)

  This book provides a comprehensive list of terms and definitions used throughout the IBM Communications Server for Linux library.

Systems Network Architecture (SNA) Publications

The following books contain information about SNA networks:
- *Systems Network Architecture: Format and Protocol Reference Manual—Architecture Logic for LU Type 6.2* (SC30-3269)
- *Systems Network Architecture: Formats* (GA27-3136)
- *Systems Network Architecture: Guide to SNA Publications* (GC30-3438)
- *Systems Network Architecture: Network Product Formats* (LY43-0081)
- *Systems Network Architecture: Technical Overview* (GC30-3073)
- *Systems Network Architecture: APPN Architecture Reference* (SC30-3422)
- *Systems Network Architecture: Sessions between Logical Units* (GC20-1868)
- *Systems Network Architecture: LU 6.2 Reference—Peer Protocols* (SC31-6808)
- *Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2* (GC30-3084)
- *Systems Network Architecture: 3270 Datastream Programmer's Reference* (GA23-0059)
- *Networking Blueprint Executive Overview* (GC31-7057)
- *Systems Network Architecture: Management Services Reference* (SC30-3346)

APPC Publications

The following books contain information about Advanced Program-to-Program Communication (APPC):
- *APPC Application Suite V1 User's Guide* (SC31-6532)
- *APPC Application Suite V1 Administration* (SC31-6533)
- *APPC Application Suite V1 Programming* (SC31-6534)

- *APPC Application Suite V1 Online Product Library* (SK2T-2680)
- *APPC Application Suite Licensed Program Specifications* (GC31-6535)
- *z/OS V1R2.0 Communications Server: APPC Application Suite User's Guide* (SC31-8809)

Programming Publications

The following books contain information about programming:

- *Common Programming Interface Communications CPI-C Reference* (SC26-4399)
- *Communications Server for OS/2 Version 4 Application Programming Guide* (SC31-8152)

Index

Communicating your comments to IBM

If you especially like or dislike anything about this document, use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this document. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Send your comments to us in any of the following ways:
- To send comments by FAX, use this number: 1+919-254-1258
- To send comments electronically, use this address: comsvrcf@us.ibm.com
- To send comments by post, use this address:

  International Business Machines Corporation
  Attn: z/OS® Communications Server Information Development
  P.O. Box 12195, 3039 Cornwallis Road
  Department AKCA, Building 501
  Research Triangle Park, North Carolina 27709-2195

Make sure to include the following in your note:
- Title and publication number of this document
- Page number or topic to which your comment applies.

IBM®

Product Number: 5725-H32

Printed in USA