Communications Server
for Windows®

Quick Beginnings

Version 6.1.3

GC31-8424-05

Communications Server
for Windows®

Quick Beginnings

Version 6.1.3

GC31-8424-05

Note

Before using this information and the product it supports, be sure to read the information under Appendix C, "Notices," on page 95.

Sixth Edition (October 2007)

This edition applies to Version 6.1.3 of Communications Server for Windows® and to all subsequent releases and modifications until otherwise indicated in new editions or technical newsletters.

© Copyright International Business Machines Corporation 1997, 2007. All rights reserved.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

© Copyright IBM Corp. 1997, 2007

Figures

© Copyright IBM Corp. 1997, 2007

Tables

© Copyright IBM Corp. 1997, 2007

About this document

IBM® Communications Server for Windows® is a communications services platform. This platform provides a wide range of services for Windows workstations that communicate with host computers and with other workstations. Communications Server users can choose from among a variety of remote connectivity options.

This document describes the IBM Communications Server for Windows and explains how to plan for, install, and use it. The term *Communications Server* refers to the Windows version of the product, unless otherwise specified.

How this document is organized

If you want to...	Refer to...
Read about Communications Server.	Chapter 1, "About IBM® Communications Server for Windows," on page 1
Plan for the installation of Communications Server.	Chapter 2, "Planning for the Installation of Communications Server," on page 19
Install Communications Server.	Chapter 3, "Installing Communications Server," on page 21
Configure Communications Server.	Chapter 4, "Configuring Communications Server for Windows," on page 43
Start using Communications Server.	Chapter 5, "Using Communications Server," on page 65
Display diagnostic information.	Chapter 6, "Problem Determination and Reporting," on page 79

How to contact IBM service

For immediate assistance, go to the following Web site: http://www.ibm.com/software/network/commserver/windows/support

Most problems can be resolved at this Web site, where you can submit questions and problem reports electronically, as well as access a variety of diagnosis information.

For telephone assistance in problem diagnosis and resolution (in the United States or Puerto Rico), call the IBM Software Support Center anytime (1-800-IBM-SERV). You will receive a return call within 8 business hours (Monday – Friday, 8:00 a.m. – 5:00 p.m., local customer time).

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized IBM supplier.

Conventions and terminology used in this document

Knowing the conventions used in this book will help you use it more efficiently.

- **Boldface type** indicates:

© Copyright IBM Corp. 1997, 2007

  - Verbs, functions, and parameters that you can use in a program or at a command prompt. These values are case sensitive and should be entered exactly as they appear in the text.
  - The names of window controls, such as lists, check boxes, entry fields, push buttons, and menu choices.
- *Italics type* indicates:
  - A variable for which you supply a value
  - Book titles
  - A letter that is being used as a letter or a word that is being used as a word. For example, when you see an *a*, make sure it is not supposed to be an *an*.
- ***Bold italics type*** is used to emphasize a word.
- UPPERCASE indicates constants, file names, keywords, and options that you can use in a program or at a command prompt. You can enter these values in uppercase or lowercase.
- `Example type` indicates information that you are instructed to type at a command prompt or in a window.
- Double quotation marks indicate messages you see in a window. An example of this would be the messages that appear in the operator information area (OIA) of an emulator session.
- Binary numbers are represented as B'xxxx xxxx' or B'x' except in certain instances where they are represented with text ("A value of binary xxxx xxxx is...").
- Bit positions start with 0 at the right position (least significant bit).
- Decimal numbers over 4 digits are represented in metric style. A space is used rather than a comma to separate groups of 3 digits. For example, the number sixteen thousand, one hundred forty-seven is written 16 147.
- Hexadecimal numbers are represented in text as hex xxxx or X'xxxx' ("The address of the adjacent node is hex 5D, which is specified as X'5D'.")

This book also uses *icons* (pictures) in the text to help you find different types of information.

This icon represents a note, important information that can affect the operation of Communications Server or the completion of a task.

This icon represents a hint or additional information that can help you complete a task.

For definitions of the terms and abbreviations used in this document, you can view the latest IBM terminology at the IBM Terminology Web site.

Prerequisite and related information

This section describes the sources of information that can be useful when you are using Communications Server or related products.

Required information

Communications Server Library

The following documents are provided as Adobe Acrobat Portable Document Format (PDF) files on the Communications Server for Windows CD-ROM:

Table 1. Communications Server for Windows Library

Order Number	Title	Description
GC31-8424	*Quick Beginnings*	Provides an overview of Communications Server and the access features and emulator provided with it. Also gives installation and configuration information.
SC31-8425	*Client/Server Communications Programming*	Describes the advanced program-to-program communications (APPC) application programming interface (API) support provided by Communications Server.
SC31-8655	*Configuration File Reference*	Describes how to use an ASCII editor to copy or change configuration files. This book lists all of the configuration file keywords and their valid values.
SC31-8656	*Network Administration Guide*	Provides information for the Network Administrator's use in planning and maintaining networks that use Communications Server.
SC31-8426	*System Management Programming*	Describes the usage of the Node Operator Facility (NOF) API messages for developing programs using Communications Server.

For information about ordering publications listed in this document, contact your IBM authorized dealer or marketing representative. In the United States, you can also order publications by dialing **1-800-879-2755**. In Canada, you can also order publications by dialing **1-800-IBM-4YOU** (**426-4968**).

You can view the books directly from the CD-ROM. The .PDF files are located as follows:

Client/Server Communications Programming

auxiliaryfiles\sdk\pubs\<*lang***>\cscsp.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

Configuration File Reference

auxiliaryfiles\doc\<*lang***>\cscfr.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

Network Administration Guide

auxiliaryfiles\doc\<*lang***\csnag.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

Quick Beginnings

auxiliaryfiles\doc\<*lang***\csqkb.pdf**

where <***lang***> is one of the following languages:

- chs
- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

SNA Management Services formats

auxiliaryfiles\sdk\pubs\snams.pdf

SNA Format Reference

auxiliaryfiles\sdk\pubs\snafmt.pdf

SNA Format Reference

auxiliaryfiles\doc\snafmt.pdf

System Management Programming

auxiliaryfiles\sdk\pubs\<*lang***>\cssmp.pdf**

where <***lang***> is one of the following languages:

- chs

- cht
- deu
- enu
- esp
- fra
- ita
- jpn
- kor
- ptb

CPIC Reference

auxiliaryfiles\sdk\pubs\cpicref.pdf

Related Information

Additional information is contained in the following related publications:

- *Common Programming Interface Communications Reference* SC26–4399

  Describes CPI Communications. It is intended for programmers who want to write applications that use communications products supporting CPI Communications. Available on the Communications Server CD-ROM in HTML and PDF formats.

- *IBM Dictionary of Computing*

  Defines technical terms used in the documentation for IBM products. Available on the Web at http://www.ibm.com/networking/nsg/nsgmain.htm

- *Systems Network Architecture Formats* GA27–3136

  Describes SNA formats used between subarea nodes and peripheral nodes, and between nodes implementing APPN or low-entry networking (LEN) protocols. Available on the Communications Server CD-ROM in PDF format.

- *Systems Network Management Services Reference* GC31–8302

  Describes SNA Management Services formats used between nodes in subarea networks, and between nodes implementing APPN or low-entry networking (LEN) protocols, or both. Available on the Communications Server CD-ROM in PDF format.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this document, do one of the following:

- Send your comments by e-mail to comsvrcf@us.ibm.com.
- If you prefer to send comments by post, use this address:

```
International Business Machines Corporation
Attn: IBM Communications Server for Windows Information Development
P.O. Box 12195, 3039 Cornwallis Road
Department AKCA, Building 501
Research Triangle Park, North Carolina 27709-2195
```

Be sure to include the name of the document, the part number of the document, the version of Communications Server for Windows, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

Summary of changes

Summary of changes
for GC31-8424-06
V6.1.3

This document contains information previously presented in GC31-8424-05, which supports V6.1.2.

New information

- Server and Remote API Client operating system support on the following platforms:
  - Windows Vista (32-bit only)
- Windows NT is no longer supported on the server and clients
- SNA API Client for the following operating systems:
  - Windows Server 2003 (64-bit), Windows XP 64-bit)
  - Windows Vista (32-bit and 64-bit)
- Enterprise Extender supports IPv6
- Easy installation using Microsoft Software Installer (MSI)
- Connection Network Reachability Awareness support
- CPI-C Communication Server compatibility tool for easier migration
- Synchronous SLI Close for Microsoft compatibility allows you to migrate applications from a Microsoft HIS Server to Communications Server for Windows.

© Copyright IBM Corp. 1997, 2007

Chapter 1. About IBM® Communications Server for Windows

Today's personal computers offer new ways of networking directly from your desktop. For example, you can communicate and share resources with someone in the next room, the next city, or on another continent. At the same time, an application running on your workstation can be accessing data from a host computer while distributing data to another workstation on a local area network (LAN). You can even have a conference from your personal computer with live video and interactive data exchanges across a network. Distributed processing and client/server applications can also share the personal computer with terminal emulator programs that connect you to applications running on a host computer. These are only a few of the applications of personal networking from your desktop workstation.

IBM's Communications Server for Windows provides an industrial-strength networking solution for your workstation. Whether it's for host terminal emulation, client/server and distributed applications, or connectivity across local and wide area networks (LANs and WANs), Communications Server offers a robust set of communications, networking, and system management features.

For true networking flexibility, a wide range of connectivity services and options are provided. With Communications Server, workstations and gateways can communicate using communication protocols such as TCP/IP and SNA. Communication occurs over a variety of data link controls (DLCs). Mobile users can directly access their host system or another Communications Server through public networks.

The versatility of Communications Server extends to the types of applications that can be supported. Communications Server supports a wide variety of application programming interfaces (APIs) and protocols that are ideal for client/server applications and distributed processing. Communications Server also protects your investment in applications by providing compatible APIs for clients and servers. Programs using these APIs can run on any node in the network, whether it is a client or server.

The Common Programming Interface for Communications (CPI-C) and advanced program-to-program communication (APPC) support makes Communications Server the ideal communications platform for peer application environments. The Logical Unit APIs (LUA) for applications that need access to LU 0, 1, 2, and 3 data streams are available for access to many host applications. Refer to the Communications Server Software Development Kit (SDK) for header files, samples, and documentation regarding the APIs.

Advanced Peer-to-Peer Networking® (APPN) adds additional power to the basic SNA functions supported by Communications Server. APPN is a protocol that can greatly enhance the management of an SNA network. This is accomplished through reduced configuration requirements, dynamic directory searches, route calculation capabilities, and intermediate session routing.

Communications Server contains a powerful SNA gateway facility. The gateway allows workstations on a local area network to connect to host computers through one set of adapters and physical connections. The Communications Server gateway supports various workstations. Remote workstations can also dial in to this

© Copyright IBM Corp. 1997, 2007

gateway and then use a common high-speed link to the host. The gateway performs the function of an LU concentrator making LU resources from any number of host and host PUs available to any number of downstream client workstations over any combination of communications media supported by Communications Server. The link to the upstream host can be either traditional SNA subarea or an APPN DLUR link.

Communications Server also delivers solutions to address multiprotocol challenges with its AnyNet®, SNA API Clients, TN5250 server, and TN3270E server offerings.

AnyNet software enables application programs to communicate over different transport networks and across interconnected networks. Using AnyNet, you can reduce the number of transport networks and therefore reduce operational complexity. These benefits are gained without modification to your existing application programs or hardware.

Communications Server delivers AnyNet SNA over TCP/IP access node and gateway. Thus, you can combine SNA and TCP/IP networks without losing application support.

Another way to combine SNA and TCP/IP is by using the SNA API Clients that are shipped with Communications Server. Communications Server establishes and manages a session as if the origin were the local node, and then forwards the necessary application data over TCP/IP to the SNA API Clients. SNA API Clients are supported on Windows.

The TN3270E server enables TN3270- and TN3270E-compliant clients to gain access to host applications. By configuring a TN3270E server, you can exchange data between TCP/IP workstations running TN3270E applications and a Systems Network Architecture (SNA) mainframe host computer. The TN3270E server function supports the same upstream DLCs, PU and LU support, and pooling as the SNA gateway.

The TN5250 server enables TN5250 clients on a TCP/IP network to gain access to applications on an SNA-connected iSeries™.

Communications Server includes several tools to simplify the administration of SNA networks. These include system and network alerts, message and error logs, trace files, and programming capabilities that allow you to configure and manage networks and build sophisticated management programs. Complementary network and system management products can access all of these facilities.

The Remote Administration feature is a tool that gives network administrators the ability to manage Communications Server over the Internet or your company's intranet using a node operations panel.

You can get up-to-the-minute information about the IBM family of products from the Communications Server home page: `http://www.ibm.com/software/network/commserver`. This page includes information and links to headline information, specification sheets, frequently asked questions and answers, maintenance and support information, and more.

Features and Functions of Communications Server

The Communications Server package contains the following product features and functions:

- APPN network node and end node support
- High-Performance Routing (HPR)
  - Rapid Transport Protocol (RTP)
  - Automatic Network Routing (ANR)
  - Enterprise Extender (HPR over IPv4 and IPv6)
- SNA data compression
- Discovery of service providers
- Dependent LU Requester (DLUR)
- SNA gateway
- TN3270E Server
- TN5250 Server
- iSeries OLE DB Provider
- SNA API Client services
- AnyNet SNA over TCP/IP (access node and gateway)
- Local and remote configuration and administration support
- Load balancing
- 32-bit application programming interfaces (APIs)
  - CPI-C
  - APPC
  - LUA RUI
  - LUA SLI
  - Management Services API
  - Common Services API
  - Node Operations API
- Data security
- LAN support
- Local and wide area connectivity support
- OEM communication adapter support

A machine where Communications Server for Windows is installed is often termed a **node**. A Communications Server node can be configured to enable any or all of the features listed above.

Description of the Communications Server Features and Functions

This section gives more information about each of the Communications Server features and functions.

Advanced Peer-to-Peer Networking (APPN) Network and End Node Support

Advanced Peer-to-Peer Networking (APPN) is a networking extension to APPC which simplifies configuration and enhances management of a group of workstations using APPC or CPI-C transaction programs. An APPN network is made up of network nodes and end nodes. The APPN network nodes provide directory services, route selection, and management services to end nodes. The end nodes need only know the location of the network node to communicate with any node in the APPN network. SNA discovery support further simplifies configuration by automatically finding network nodes for the end nodes on a LAN network.

With APPN, you can:

- Add, delete, or move nodes within the network with limited system definition at the affected node, and no other definition at other nodes
- Use defaults for reducing required system definition
- Significantly improve the performance of communications between APPC and CPI-C applications, especially in a LAN environment
- Use the NOF API to automate configuration changes and add network management capabilities

High-Performance Routing

Communications Server supports high-performance routing (HPR), which increases data routing performance and reliability of APPN networking. HPR provides nondisruptive rerouting around network outages, efficient selective retransmission, and end-to-end data integrity and congestion control. Communications Server supports both:

Rapid Transport Protocol (RTP)
This allows a node to be the initiation or termination point of an HPR conversation.

Automatic Network Routing (ANR)
This allows a node to act as an intermediary along the path of an HPR conversation.

Communications Server supports HPR routing over IP, LAN, synchronous data link control (SDLC), and X.25 connections. You can use HPR in host or non-host-oriented networks.

SNA Data Compression

Data compression at the session level increases throughput for large amounts of data across communication links, resulting in the following benefits:

- Enhanced data throughput on low-speed lines
- Reduced costs on high-cost lines
- Faster response times, resulting in productivity improvements

SNA data compression is compatible with the zSeries® and iSeries implementations and can be used with all LU types.

Discovery of Service Providers

Discovery is a LAN address resolution protocol that can be used by a node on the LAN to find another node that matches given search criteria. By adjusting the search parameter, a node can search for APPN network nodes, nodes that provide SNA boundary function, iSeries, SNA gateways, or user-defined classes of service. A Communications Server for Windows server can respond to requests from clients as a network node server, a PU 2.0 gateway, or as a user-defined class of service. A Communications Server can also use discovery to find APPN nodes and SNA gateways.

Dependent LU Requester (DLUR)

Communications Server enables the support in VTAM for dependent LUs through APPN networks and combined subarea and APPN networks. The dependent LU server function (in VTAM) provides dependent secondary logical unit (SLU) support by establishing an LU 6.2 session using the SNASVRMGR mode between a dependent LU requester node (DLUR), and a dependent LU server node (DLUS).

A DLUR is an APPN end node or network node that uses dependent LUs, but requests that a DLUS provide the system services control point (SSCP) for those dependent LUs through an APPN network. A DLUS controls conversion from a subarea environment to an APPN environment, allowing you to maintain central management of remote dependent LUs while benefiting from an APPN network.

DLUR allows dependent LUs (LU 0, 1, 2, 3, and dependent LU 6.2) to benefit from an APPN network. It supports dynamic and multiple paths through the network and eliminates the need for dependent LUs (or their gateway) to be adjacent to the VTAM host.

SNA Gateway Support

Communications Server provides a full-function Systems Network Architecture (SNA) gateway. The gateway allows multiple LAN-attached workstations to access System/370™ or zSeries hosts through one or more physical connections to one or more hosts. This helps reduce the cost per workstation of host connections.

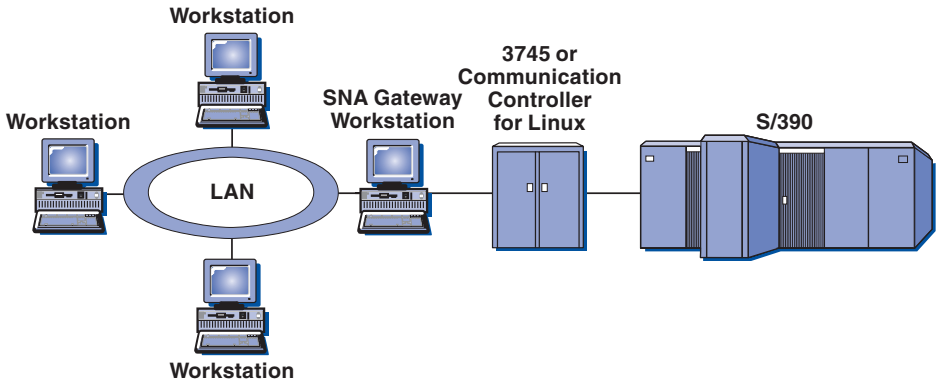

Figure 1. Example of SNA Gateway Configuration

The Communications Server gateway supports the SNA protocols LU 0, 1, 2, 3, and dependent LU 6.2 (APPC). With the AnyNet SNA over TCP/IP function, downstream workstations can now communicate with the SNA gateway over an IP network. The gateway also supports LU 0, 1, 2, or 3 to an iSeries host using SNA pass-through. The iSeries host passes the data through to a zSeries host.

A gateway can also act as a protocol converter between workstations attached to a LAN and a WAN host line.

The LUs defined in the gateway can be dedicated to a particular workstation or pooled among multiple workstations. Pooling allows workstations to share common LUs, which increases the efficiency of the LUs and reduces the configuration and startup requirements at the host. You can also define multiple LU pools, each pool associated with a specific application. And you can define common pools that are associated with multiple hosts. When a client connects to the gateway, the gateway retrieves an LU from the pool to establish a session. The LU is returned to the pool for access by other workstations when the session is ended.

In addition, an SNA gateway can support the forwarding of network management vector transports (NMVTs) between the workstations and the host.

Each host views the SNA gateway as an SNA PU 2 node, supporting one or more LUs per workstation. As far as the host is concerned, all LUs belong to the SNA

gateway PU. The SNA gateway can have multiple host connections simultaneously and can direct different workstation sessions to specific hosts.

To the supported workstations, the SNA gateway looks like an SNA PU 4 communications controller and forwards such host requests as BIND and UNBIND. The workstation LUs are not aware of the SNA gateway. The SNA gateway, however, is aware of all LUs at the workstations.

Downstream applications using standard SNA connectivity protocols for LU 0, 1, 2, and 3 and dependent 6.2, and communicating through an SNA gateway to a host, are supported by Communications Server. Table 2 summarizes the SNA gateway features.

Table 2. SNA Gateway Summary

Feature	Description
Active workstations	254 (LAN) per adapter 128 (X.25)
DLCs	AnyNet (SNA over TCP/IP) LAN (Any NDIS** compliant network adapter) X.25 SDLC OEM adapters MPC Channel Enterprise Extender (upstream only, requires DLUR)
Downstream workstations	Any product that supports standard SNA connectivity protocols for LU 0, 1, 2, 3, and 6.2.
Dynamic additions and changes	Yes
Implicit workstation support	Yes
LU pooling	Yes
Maximum number of LUs	254 per PU; no limit on the number of PUs
Mode of operation	Multiple downstream PUs (not apparent to the host) PUs not visible to the host (except when through DLUR)
Multiple PU support	Yes
Segmenting support	Yes
Supported LU types	LU 0, 1, 2, 3, and dependent 6.2

TN3270E Server

The TN3270E server function enables you to configure your network as shown in Figure 2 on page 7.

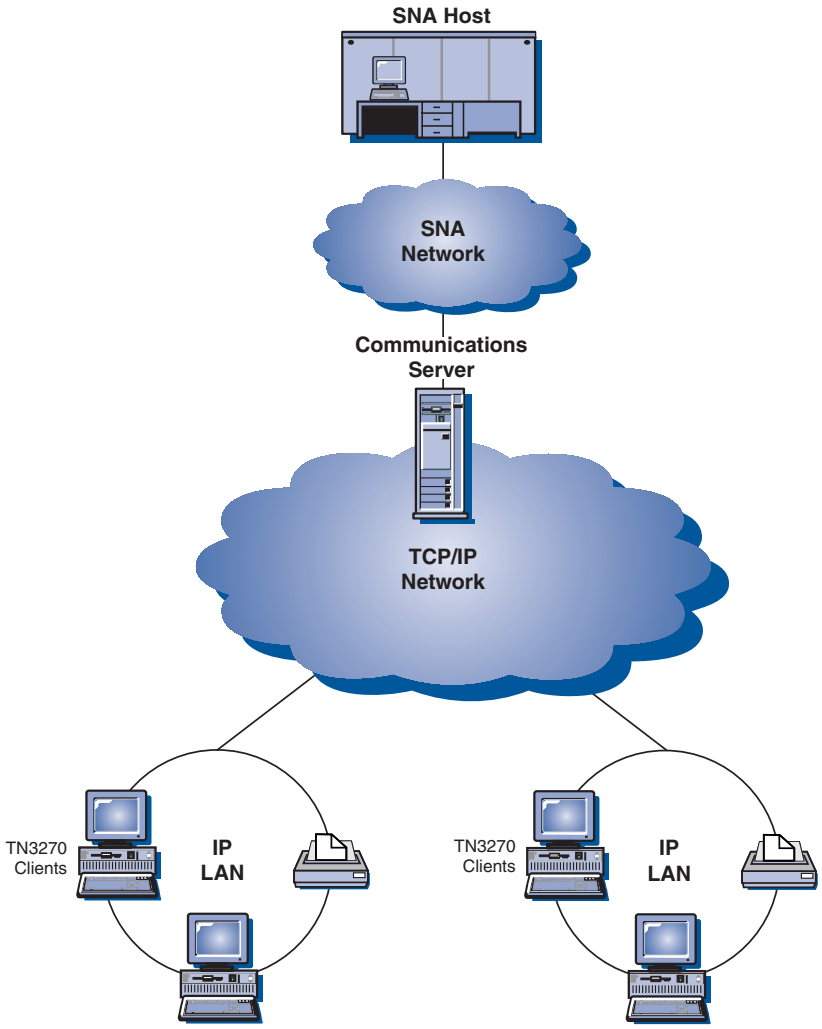

Figure 2. TN3270E Server with Wide Area TCP/IP Network

The TN3270E server function supports:

- Terminal emulation

  TN3270E server supports LU 2, which enables users to run interactive 3270 display application programs.

- Host print

  The Telnet 3270 standard extensions (TN3270E) enable users to print from host applications to printers attached to their workstation. These printers can be locally or network attached. Print sessions can be LU 1 or LU 3.

  TN3270E server implements the protocols outlined in RFC 1576, RFC 1646, and RFC 1647, enabling the server to pass LU 1 and LU 3 session data to TN3270E-enabled clients, wait for client confirmation of the print request, and respond to the host. RFC 1522 contention resolution negotiation is also supported for TN3270E clients.

- Load balancing

  You can load balance TN3270E connections. Refer to the *Network Administration Guide* for more information about load balancing.

- Client filtering

TN3270E server enables the user to specify at the server the clients, based on IP address or hostname, that can access the LU names and pools that have been configured for the TN3270E server. This allows controlled access to LUs without modifying the client configurations.

- Support for security

  You can specify that a TN3270E server port is secure, indicating that SSL Version 3 is used for connections on that port to provide data encryption and server authentication using digital certificates. You can specify the level of security for connections on the port, whether client authentication should be processed, and, if so, whether a Certificate Revocation List should be checked before accepting the client's connection.

- Multiport support

  The TN3270E Server allows the configuration of one or more ports for the TN3270E Server to listen for incoming connections. Each port is configured with its own parameters such as whether or not connections on the port are secure and what default pool should be used for incoming sessions.

- LU_TAKEOVER

  Enables the user to specify that LU takeover processing is used for unused LU resources at the server instead of keepalive processing.

- Response handling

  TN3270E-enabled clients can send both positive and negative responses, which TN3270E Server sends on to the host. TN3270E server generates responses for standard TN3270 clients.

- ATTN and SYSREQ key handling

  TN3270E server can convert and forward information to the host when the client sends an ATTN or SYSREQ key. Although clients that are not TN3270E-enabled have no explicit definition for ATTN and SYSREQ, the TN3270E server uses the following Telnet commands to implement these functions:

Table 3. Command Equivalents

Telnet	TN3270E	Standard TN3270
IP	ATTN	SYSREQ
AO	SYSREQ	SYSREQ
BREAK	N/A	ATTN

- LU classes

  Communications Server categorizes user connections with LU classes. Classes consist of LUs configured with common characteristics; those that require a specific host connection, for example. This simplifies user access, groups users by application needs, and maximizes host resources.

TN3270E server supports both standard and extended Telnet 3270. Typical client programs emulate a 3270 display. Clients that support the TN3270E protocol can emulate LU 1 and LU 3 printers.

TN5250 Server

The TN5250 server function enables TN5250 clients on TCP/IP networks to communicate with iSeries hosts on SNA networks. It enables you to configure your network as shown in Figure 3 on page 9.

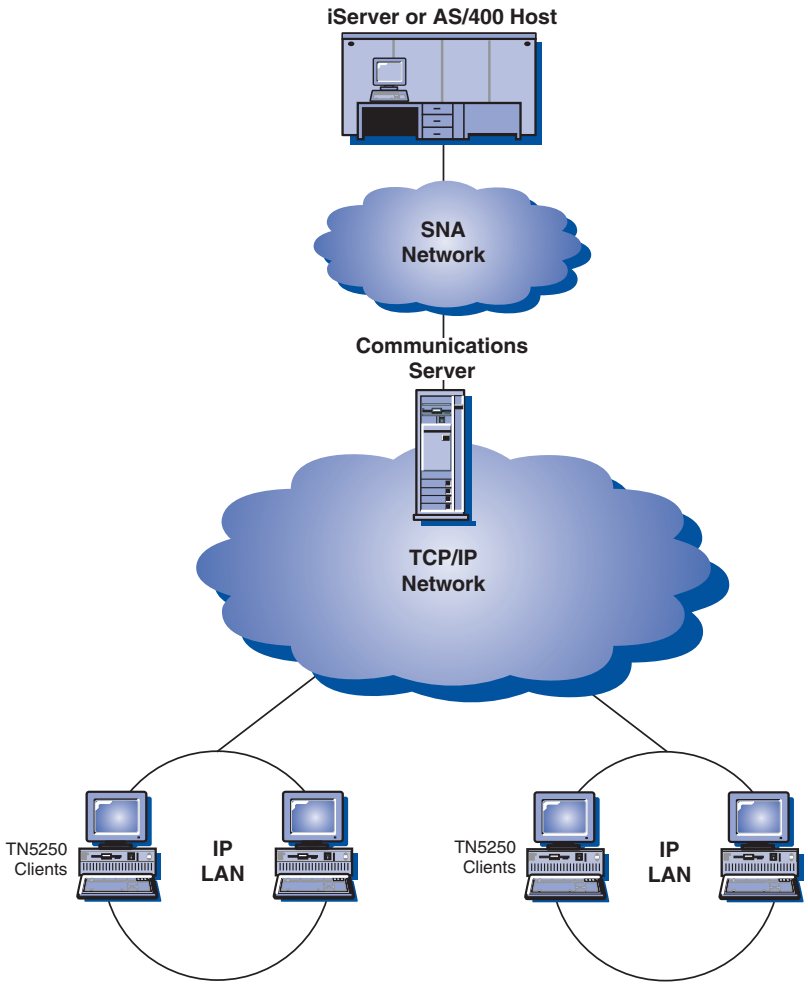

Figure 3. TN5250 Server with Wide Area TCP/IP Network

The TN5250 Server function implements the protocols outlined in RFC 1205. It supports:

- Terminal emulation

  The server provides a gateway between downstream IP-connected RFC 1205–compliant clients (TN5250 clients) and one or more upstream SNA-connected iSeries. SNA connectivity between the server and the iSeries is provided using the 5250 Display Station Pass-through datastream on LU 6.2 sessions.

- Client filtering

  TN5250 server enables the user to specify at the server the clients, based on IP address or hostname, that can access the iSeries servers that have been configured for the TN5250 server. This allows controlled access to iSeries without modifying the client configurations.

- Support for dynamic local LUs

  This support enables you to exceed the 512 maximum session limit between a single LU and the iSeries by using additional dynamically defined local LUs.

- Support for multiple iSeries

  You can enable the server to connect to multiple iSeries by configuring the server to listen on more than one IP port and specifying the iSeries that

corresponds to each port. The client specifies a port to be used for the connection, using a common configuration parameter. The server then uses the port value to identify the specific iSeries. Clients can connect to multiple hosts by specifying different ports for each emulator session.

- Support for security

  You can specify that a TN5250 server port is secure, indicating that SSL Version 3 is used for connections on that port to provide data encryption and server authentication using digital certificates. You can specify the level of security for connections on the port, whether client authentication should be processed, and, if so, whether a Certificate Revocation List should be checked before accepting the client's connection.

iSeries OLE DB Provider

Applications that use OLE DB or ActiveX can communicate through Communications Server for record-level access to files on iSeries hosts.

You can use the iSeries OLE DB Provider to connect Windows clients through Communications Server to one or more iSeries hosts, as shown in Figure 4. This configuration gives clients that have the iSeries OLE DB Provider installed access to specific records within files on the iSeries.

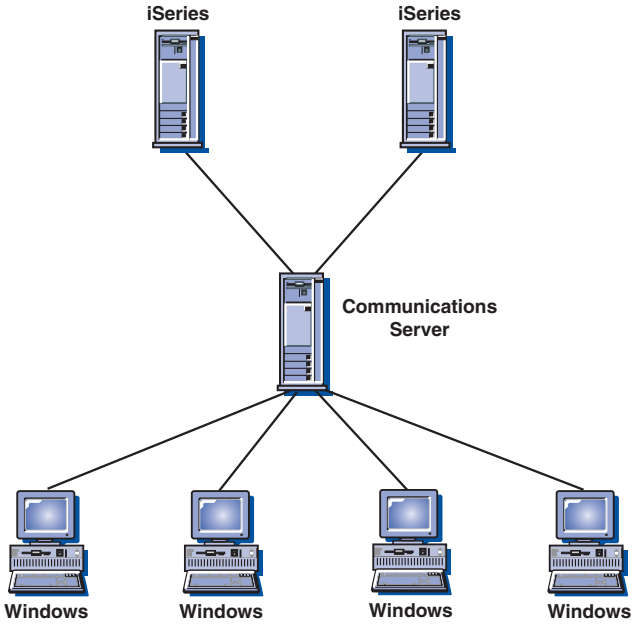

Figure 4. iSeries OLE DB Provider

In this figure, both clients have the iSeries OLE DB Provider installed.

SNA API Client Support

The Communications Server SNA API Client support allows TCP/IP clients to access SNA APIs without requiring SNA protocols to flow between the clients and the server. This allows most SNA configuration to take place at the central server.

Communications Server supports SNA API Clients on Windows as illustrated by Figure 5 on page 11.

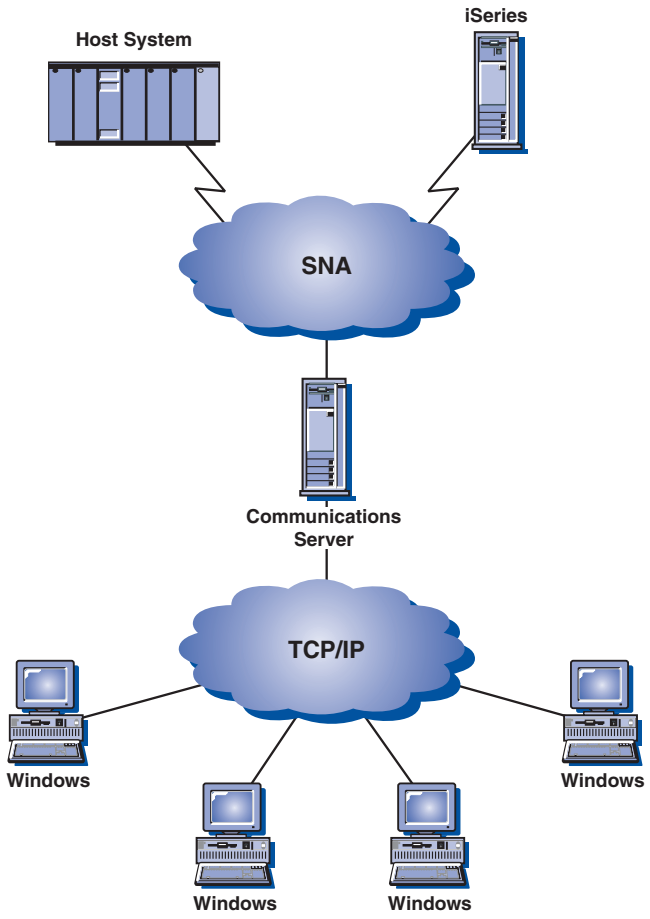

Figure 5. TCP/IP SNA API Clients

The SNA clients provide support for CPI-C APPC, EHNAPPC, LUA RUI, JCPI-C, and HACL API interfaces, while providing the actual SNA processing at the server. These clients are delivered as part of the server but are actually installed and configured at the client.

Refer to "Programming Support" on page 14 for more information about the programming support available and the Software Developers Kit.

AnyNet SNA over TCP/IP Access Node and Gateway

The AnyNet SNA over TCP/IP function in Communications Server allows SNA applications to communicate over interconnected IP and SNA networks. This is an alternative implementation to EEDLC that uses datagrams rather than HPR protocols.

The SNA over TCP/IP access node function allows SNA applications residing on an IP network to communicate. This function supports independent LU6.2 and dependent LU 0, 1, 2, 3, or 6.2 either with or without dependent LU requester (DLUR). In addition, the SNA over TCP/IP access node can be used in conjunction with SNA gateway to enable SNA gateway sessions over TCP/IP.

The SNA over TCP/IP gateway function extends the reach of SNA applications by allowing SNA applications in an SNA network to communicate with SNA applications in an IP network. The SNA over TCP/IP gateway supports independent LU 6.2 sessions.

For more information on how to configure AnyNet SNA over TCP/IP, see the online *Network Administration Guide*.

The following network configurations illustrate how SNA over TCP/IP access nodes and gateways can be used.

Running APPC or CPI-C Applications over a TCP/IP Network

Figure 6 illustrates how you can use the AnyNet SNA over TCP/IP access node function of Communications Server to enable communication between SNA applications over an IP network.

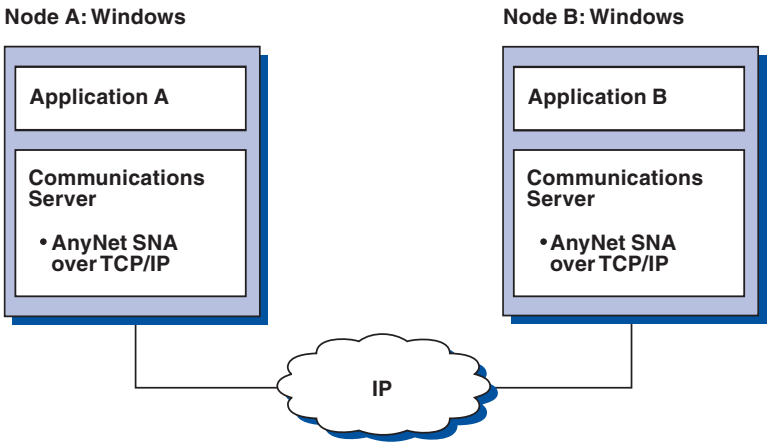

Figure 6. Running APPC or CPI-C Applications over a TCP/IP Network

Using AnyNet SNA over TCP/IP to Connect SNA Applications on Different Platforms

Figure 7 on page 13 illustrates some of the many products with AnyNet SNA over TCP/IP function. AnyNet gateways (Node D) allow SNA applications to communicate across a combination of TCP/IP and SNA networks. Note that Enterprise Extender is recommended instead of AnyNet for SNA over TCP/IP. Host systems no longer support AnyNet starting with V1R8. Communications Server for Linux® never supported it. Communications Server for Windows supports both dependent and independent LUs while Communications Server for AIX® supports only independent LUs.SNA over TCP/IP only works in IPV4 networks. Mapping SNA LUs to IPV6 addresses is not supported.

Communications Server for Windows AnyNet can be used to connect dependent LUs across an IP network without requiring DLUR at the workstations.

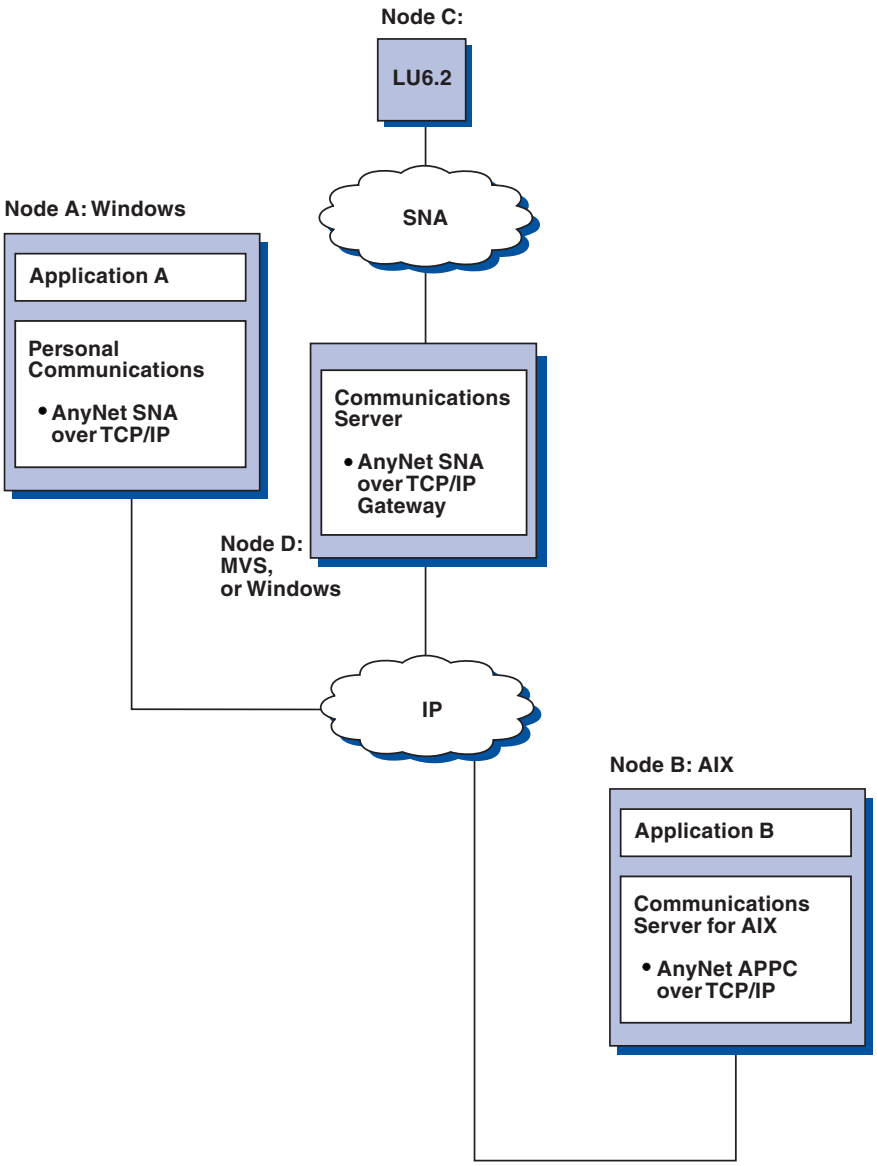

Figure 7. Using AnyNet to Connect SNA Applications on Different Platforms

Using an SNA Gateway for 3270 Emulation over a TCP/IP Network

Figure 8 on page 14 illustrates a configuration that supports dependent LU communication for downstream workstations over a single IP network. All three machines are running AnyNet SNA over TCP/IP. Node A is a dependent LU server (DLUS), and Node B is a dependent LU requester (DLUR).

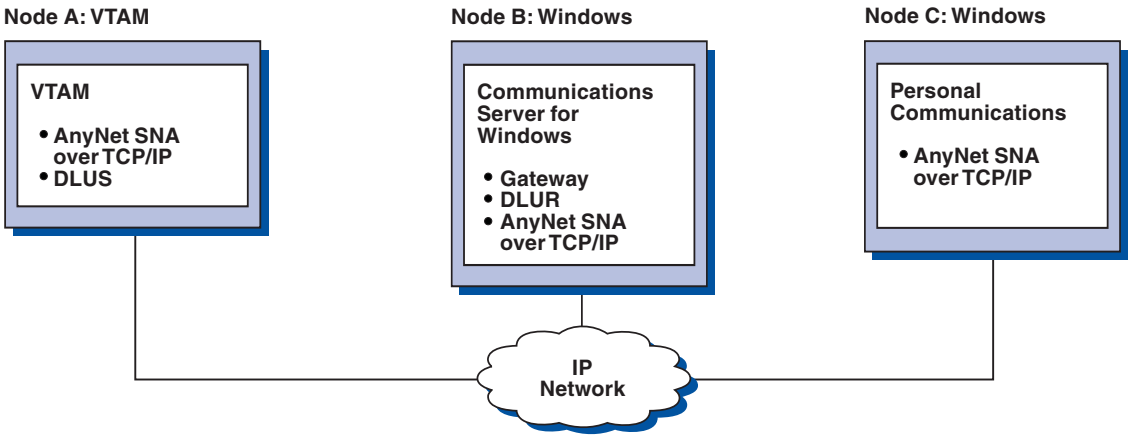

Figure 8. 3270 Emulation through an SNA Gateway over a TCP/IP Network

Local and Remote Configuration and Administration Support

Communications Server provides a graphic user interface (GUI) for entering configuration data that simplifies common configuration tasks. Local configuration is supported at both the client and server level. Remote configuration of the server is supported from Windows clients.

The **SNA Node Operations** application allows you to stop, start, and monitor resources in your network. The **SNA Node Operations** application is also supported from Windows clients. This support allows Node Operations to connect to and manage remote Communications Servers.

Communications Server includes command line utilities that enable you to perform many of the functions provided by SNA Node Operations. See "Command Line Utilities" on page 71 for more information.

Programming Support

Communications Server supports a wide range of 32-bit application programming interfaces (APIs) on the server for the application program developer. These APIs provide convenient ways for application programs to access Communications Server functions and allow applications to address the communication needs of connections to both IBM and other computers. In addition, the provided interfaces support SNA protocols so that standardization is ensured.

The APIs supported include:
- Advanced program-to-program communications (APPC)
- Common Programming Interface for Communications (CPI-C)
- Conventional LU Application Interface (LUA) RUI and SLI
- Host Access Class Libraries (HACL)
- Java™ CPI-C (JCPI-C)
- Network Operator Facility
- Management Services
- Common Services

On the clients, the Enhanced APPC (EHNAPPC) API is also provided.

The Communications Server Software Developers Tool Kit (which can be separately installed from the Communications Server CD-ROM) is also available for application developers to use. This tool kit contains samples, header files, library files, and online manuals for each of the APIs.

CPI Communications and APPC Support

Advanced program-to-program communications (APPC) supports communications between distributed processing programs, called transaction programs (TPs). APPC uses the LU 6.2 protocol for exchanging data between programs located at different logical units (LUs). In addition, APPC supports multiple concurrent links and parallel sessions. Conversation or session security between the communicating programs is also supported through APPC.

Communications Server supports APPC full duplex conversation, which greatly enhances data transmission capability and improves productivity for programmers who write conversation type applications.

A CPI-C transaction program is similar to an APPC transaction program; both types of transaction programs use APPC support. Rather than building a separate control block for each function, a CPI-C transaction program invokes each CPI-C function with a call to the function that passes the appropriate parameters on the call.

APPC is a single entry point API where complex control blocks are required to cause different functions. CPI-C provides many functions, each with its own parameters, which enhances readability and portability.

The CPI-C and APPC verbs provide similar services. However, CPI-C provides an API that is portable across multiple platforms.

Data Security

Communications Server provides basic and enhanced security support at session and conversation levels. There is security in limiting which Windows users may access SNA resources through the SNA API Clients. Conversation security includes support for password substitution. There is also enhanced LU-LU security.

Communications Server also provides session-level encryption (SLE), which enables you to encrypt either all of the data or selected data that is transferred between the workstation and the host. If you want to protect any workstation data by using encryption, the host must also be configured to use encryption.

Conformance with SNA

Communications Server supports conformance with Systems Network Architecture (SNA), the set of conventions and protocols used in IBM networks. SNA helps ensure standardization of network configurations and accurate transmission of data across networks.

An SNA network is organized as a system of *nodes* and *links*. Each node is classified according to its capabilities and the extent of control it has over other nodes in the network. The node type is not necessarily associated with a specific type of hardware, and the node's capabilities can be performed by different devices. For example, a workstation acting as a gateway can perform the same functions as a communications controller. Communications Server workstations can communicate using LAN, SDLC, X.25, and SNA over TCP/IP.

A link is the combination of the link connection (the transmission medium) and two link stations (nodes), one at each end of the link connection. A link connection can be shared among multiple links in a multipoint or LAN configuration.

LU Support

Communications Server provides support for SNA LU types 0, 1, 2, and 3, which provide communications with host applications that support devices such as:

- LU type 0, for 3650 and 4700 financial terminals
- LU type 1, for 3270 printers
- LU type 2, for 3270 interactive displays
- LU type 3, for 3270 printers

Communications Server also supports LU type 6.2 or APPC. LU type 6.2 supports communications between two programs located at type 5 subarea nodes or type 2.1 peripheral nodes, or both, and between programs and devices. Communications Server supports LU 6.2 through the APPC or CPI Communications APIs.

Management Services

Management services are functions distributed among network components to operate, manage, and control a network. This capability is based on the SNA management services architecture documented in *Systems Network Architecture Management Services Reference*.

Communications Server provides programming support that allows installation of focal point, service point, and entry point applications.

System Management

You can monitor and manage your SNA networks with several network management tools, including:

- ***Message logs and error logs***

  Communications Server writes its message-log and error-log entries to the log files maintained on each workstation.

- ***Trace files***

  Communications Server provides trace tools for problem determination.

- ***Configuration and management***

  You can use the SNA Node Operations application to help manage the resources in your network. Also, the systems management programming capabilities of Communications Server enable you to configure and manage the nodes within your SNA network. To accomplish this, Communications Server provides a subset of system management verbs that you can use to configure your node and to build sophisticated management programs for your active node.

- ***Context-sensitive helps***

System Management tools

Communications Server provides a variety of tools to assist you in network management. The tools include:

- **SNA Node Operations** is an online facility for monitoring and controlling communications resources maintained by Communications Server. It is used by individuals performing installation, testing, tuning. and special development activities of Communications Services. It can also be used to assist during problem determination. For example, you can use the **SNA Node Operations** facility to display the status of a transaction program you are having problems with.
- **Trace Services** are described in Chapter 6, “Problem Determination and Reporting,” on page 79.

- **Remote Administration Client**

Chapter 2. Planning for the Installation of Communications Server

This chapter describes the requirements for installing the elements shipped with Communications Server and gives an overview of the installation process.

Before You Start

Before you install Communications Server:

- Review the information found in the README.HTM file on the CD-ROM or the installation source directory.
- Review the available features and functions by referring to "Features and Functions of Communications Server" on page 3.
- Obtain access to the installation media (either a CD-ROM or the appropriate LAN environment).
- Obtain an administrator user ID with local authority to use when installing Communications Server.

Introduction to Installing Communications Server

Communications Server provides an interactive interface based on Microsoft Software Installer (MSI) for installing the product.

Prerequisite Software

Communication Server requires a 32-bit Microsoft Windows operating system including Windows 2000, Windows XP, Windows Server 2003 or Windows Vista. Consult the Windows documentation to determine the memory and hard disk storage requirements for your system.

TCP/IP is required for TN3270E and TN5250 servers, SNA API clients and use of AnyNet and EEDLC DLCs.

TCP/IP, IPX/SPX, or both, is required for SNA API clients, Novell NetWare for SAA® clients, and Remote Administration clients to communicate with Communications Server. If you are running IPX, you also need the following software enabled on the server:

- Gateway Service for NetWare
- SAP agent
- NWLink IPX/SPX or a compatible transport

SNA API client and Remote Administration clients require Windows 2000 or later. Note that SNA API client can run on 64-bit system but in 32-bit mode.

One of the following browsers is required to install Communications Server:

- Internet Explorer 6.0 or higher
- Mozilla 1.7 or higher
- Firefox 1.5 or higher
- SeaMonkey 1.0 or higher

© Copyright IBM Corp. 1997, 2007

Prerequisite Hardware

Communications Server Version 6.1.3 can be used on all systems that are supported by Windows 2000 or later. Consult the Windows documentation to determine the processor, memory, and hard disk storage requirements for your system. The requirements described here for Communications Server are in addition to the requirements specified for the operating system.

Configurations requiring multiple protocols, such as TN3270E, TN5250, AnyNet, or SNA API clients, require a minimum of 64MB of additional RAM on the server. More may be required based on the size of the environment. These components require both SNA and TCP/IP and a corresponding increase in memory.

Refer to the Communications Server README.HTM for more information, which can be accessed on the product CD from the start menu selections for Communications Server, or in the primary Communications Server installation directory, which by default is C:\ProgramFiles\IBM\Communications Server.

Chapter 3. Installing Communications Server

Communications Server for Windows uses Microsoft Software Installer (MSI) technology for all installation procedures. This book details how to successfully install and customize installation of Communications Server for Windows using the Microsoft Software Installer. For more information on MSI, see "Microsoft Software Installer (MSI)" on page 24.

You can link directly to the Communications Server for Windows Web site at http://www.ibm.com/software/network/commserver/windows

You will notice the following change in the installation of Communications Server for Windows Version 6.1.3.

- Microsoft Software Installer:

  If you have installed previous versions of Communications Server for Windows, you will notice significant differences in the install procedure. Communications Server for Windows Version 6.1.3 now utilizes the Microsoft Software Installer. For more information about Windows Installer see "Microsoft Software Installer (MSI)" on page 24.

Viewing documentation

The Communications Server for Windows documentation is included on the CD-ROM. Documentation can be accessed directly from the Communications Server for Windows CD-ROM or from the Common Launchpad welcome panel.

To view Communications Server for Windows, select **View Documentation** from the main panel of the Common Launchpad on the CD-ROM.

Note: You can copy the book files from the CD-ROM to a local or network drive to view at a later time.

The following sections discuss getting help when you are installing, configuring, or using Communications Server for Windows products.

Planning to install Communications Server for Windows

This section describes the companion products provided with Communications Server for Windows and topics that should be considered before installing Communications Server for Windows Version 6.1.3.

Communications Server for Windows products

The Communications Server for Windows consists of the following 3 products:

- Server
- Remote Administration Client
- SNA API Client

The server cannot be on the same machine as the clients. However, both clients can reside on the same machine.

© Copyright IBM Corp. 1997, 2007

National Language Support (NLS) considerations

Communications Server for Windows provides a multi-language CD-ROM that contains all supported languages.

This document describes the CD-ROM for the multiple language support version. For information on multiple language support directories and language codes, see "National Language Support (NLS) Abbreviations and Language Codes" on page 42.

Installing Communications Server for Windows

Communications Server for Windows provides the following installation options:

- Installation to a local hard drive, including installing to run from source, where source medium is a CD-ROM. See "Installing to a Local Hard Drive" on page 25 for more information.
- Administrative installation, including installing or running from source, where source medium is a network server. See "Administrative installation" on page 34 for more information.
- Remote installation using Microsoft Systems Management Server (SMS).
- Remote installation using Tivoli® Software Distribution. See "Considerations for Tivoli Software Distribution Support" on page 36, for more information.

Communications Server for Windows also offers the ability to customize the installation procedure. For information on customizing with initialization file processing, including performing silent installations, see "Installing Communications Server for Windows Using an Initialization (response) File" on page 30.

This section describes the general installation procedure for all options and the specific procedure for installing to a local hard drive. Typical and custom setups are described in this section, as well as silent installation options. Additionally, this section provides an introduction to the Microsoft Software Installer.

Common Launchpad

Use Common Launchpad (launchpad.exe) as the front end GUI to install Communications Server for Windows. Running launchpad.exe is the recommended path for installing the product. However, to bypass Common Launchpad, run setup.exe on the CD under the following directories:

- RemoteAdminClient (for the Remote Administrative Client)
- Server (for the Communications Server)
- SNAAPIClient (for the SNA API Client)

You must bypass Common Launchpad when using an initialization file to install (see "Installing Communications Server for Windows Using an Initialization (response) File" on page 30).

Hints when running Common Launchpad

Depending on how your machine is set up, you may encounter some problems when trying to run the launchpad. The following are common problems with their solutions:

- Launchpad failure issues **undefined** error in pop-up

  On some machines, Microsoft Windows has problems resolving short path names for directories with special characters (such as underscores).

If you copy the Communications Server image from the CD onto a machine and receive a pop-up as in Figure 9, followed by only a blank screen, this is likely that problem. To resolve the problem, copy the image to a directory with a simpler name.

Figure 9. Launchpad failure

Notice the "~4#" in Figure 9. In this case, that is the problem. Issue the **dir /x** command from a DOS prompt to see the short name of your directory. Avoid paths that contain *~x#* (where *x* is a character).

- Internet Explorer Script Error occurs on initialization of IExplore.html

  If urlmon.dll is not registered properly on your system, you may experience the error displayed in Figure 10. Follow these steps to resolve the problem:
  - Select Start->Run->Regsvr32 urlmon.dll.
  - **DllRegisterServer in urlmon.dll succeeded** message will appear.
  - For more details, see : http://support.microsoft.com/kb/306831

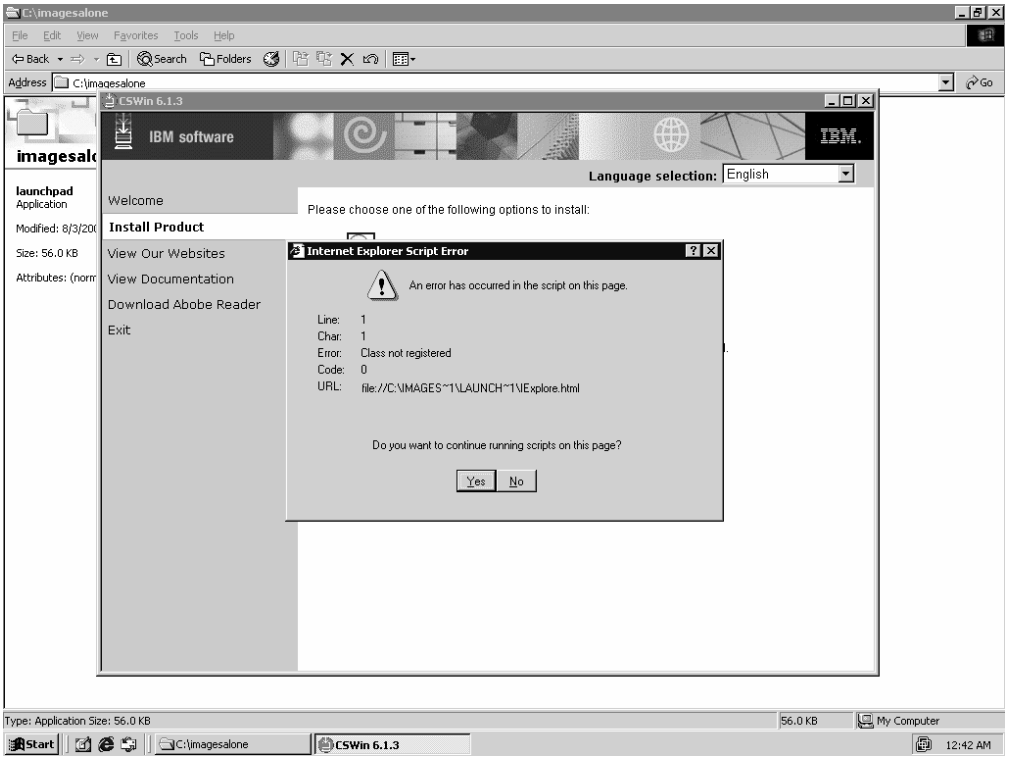

Figure 10. Internet Explorer script error

- Vista launchpad pop-up on reboot

On Vista, the Restart Manager may try to initialize launchpad after you reboot. This is a known limitation and does not cause problems. Closing the launchpad Browser window before rebooting may eliminate the pop-up. The symptom is a pop-up similar to the following:

```
This launchpad is not intended to run on the current platform.
Check the product documentation or contact your vendor for more
information about supported platforms.
Information about your machine
Operating System (top.OS) =
Operating System Type (top.OSTYPE) = windows
Processor Architecture (top.ARCHITECTURE) =
Browser/version (top.BROWSER/top.BROWSERVERSION) = IExplore/7
Locale (top.LOCALE) = en
Launchpad compatibility version =

Compatibility checks defined for this launchpad
version = 6\.1\.3
top.OS = Windows.*
top.ARCHITECTURE = x86|AMD64|IA64
top.OSTYPE = windows
```

Refer to the following link for information regarding the Windows Vista Restart Manager: http://msdn2.microsoft.com/en_us/library/aa373654.aspx

Microsoft Software Installer (MSI)

Communications Server for Windows utilizes the Microsoft Software Installer. When the Communications Server for Windows CD-ROM is first run, it examines the target system and, if necessary, automatically installs the proper version of the Microsoft Software Installer.

For each of the three products, there is a setup.exe that is the bootstrap loader that calls the Microsoft Software Installer (msiexec.exe) and launches the installation dialogs. For a detailed description of the Microsoft Software Installer, refer to the Microsoft Software Installer SDK available online at http://www.msdn.microsoft.com. For more information on setup.exe, see "InstallShield Professional for Window Installer (IPWI) command-line parameters" on page 39.

Note: The following should be taken into account when installing Communications Server for Windows:

- In some cases, installation of Microsoft Software Installer triggers a reboot of the system. If you are required to reboot, upon subsequent startup you are taken immediately back to Microsoft Software Installer to continue installation of Communications Server for Windows.
- After Microsoft Software Installer has been successfully invoked, if the installation of Communications Server for Windows fails or is cancelled by the user, Microsoft Software Installer rolls back all partially installed Communications Server for Windows files and returns the system to its original state.You should not attempt to cancel the install once it has started. Cancellation can cause problems with future installs or uninstalls.

  Note: Start menu icons from previous versions of Communications Server for Windows are not rolled back if installation fails. If this occurs, Communications Server for Windows start menu icons will no longer be available.
- You must be a member of the Administrator's group to perform these installations.

- Before you begin installation, make sure all other applications are stopped. If you are reinstalling or are upgrading Communications Server for Windows, make sure that Communications Server for Windows is not running before you start setup.
- If you are upgrading from a version that is earlier than v6.1.3, you must first uninstall the earlier version of Communications Server for Windows.

Note: The following should be taken into account when uninstalling Communications Server for Windows:

- The key files that are installed for IBM Communications Server are marked as shared files. Therefore, a reference count (a tally maintained for each shared file under HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\SharedDLLs) is created in the registry, and incremented. If you have files that do not go away when you uninstall, it can be the result of a count that was incremented during a previous failed install or uninstall. If this is the case, remove the files and update the registry. Do not update the registry if you are a general user and unfamiliar with the procedure.
- You should not attempt to cancel the uninstall once it has started. Cancellation can cause problems with future installs or uninstalls. For example, you may be unable to restore a file to it's original state before it was removed.
- If you experience problems uninstalling, run the Microsoft uninstall tool to cleanup the incomplete uninstall.

Installing to a Local Hard Drive

To install Communications Server for Windows, begin by inserting the CD-ROM into the CD-ROM drive. If AutoPlay is on, inserting the CD-ROM automatically invokes the Common Launchpad.

Note: If AutoPlay is off, do the following:

1. Click **Run...** on the Windows **Start** menu.
2. Type

   ```
   E:\launchpad.exe
   ```

   in the Command Line box (where E: is the CD-ROM drive).

From the Common Launchpad welcome dialog for Communications Server for Windows, you can choose from the following options:

- **Install Product** invokes the Microsoft Software Installer and begins the installation process.
- **View Documentation** allows you to view the books available in the Communications Server for Windows library and the Communications Server for Windows README file; *Quick Beginnings*, *Network Administration Guide* or *Configuration Reference Guide*. For more information, see "Viewing documentation" on page 21.
- **Visit Our Web site** contains links that take you directly to the Communications Server for Windows Web sites.
- **Download Adobe Reader** allows you to get a copy of Adobe Reader so you can view the pdf files.

To begin the installation of Communications Server for Windows select **Install Product** from the Common Launchpad welcome dialog.

After Common Launchpad determines the system configuration, the installation process begins.

1. The **Microsoft Software Installer welcome** dialog for the Communications Server for Windows product opens.
2. The **License Agreement** dialog opens. Click the button to accept the terms of agreement. You can print the license agreement by clicking **Print**. If you decline the license agreement, the installation process terminates. Click **Next** to continue.
3. The **Setup Type** dialog opens. You are given an option to choose a **Typical** (default) or **Custom** installation.

The following sections describe Typical and Custom installation choices.

Typical Installation

Typical installation selects all default features for installation. Features are defined as the specific functions of a program. See "Feature Selection" on page 28 for a list of default features for each of the Communications Server for Windows products. You can customize Communications Server for Windows features by selecting the custom installation option (see "Custom Installation" on page 27).

To start a typical installation, click **Next** in the installation type panel. A panel appears, indicating that Communications Server for Windows is computing the disk space requirements.

To continue with the typical installation, use the following installation procedure.

1. The **Application Data Location** dialog opens. Select from the following application data location options:
   - All users' common application data folder (All Users\Application Data)
   - Classic private directory

   Communications Server for Windows uses multiple configuration files: user-class files can be stored individually by user profile, while system-class files are stored in a common location.

   If the All Users\Application Data location is selected, the following profile paths are used:

Operating System	User-Class Directory (Current User)	System-Class Directory
Windows 2000,Windows Server 2003, Windows XP, Windows Vista	C:\Documents and Settings\All Users\Application Data\IBM\Communications Server	C:\Documents and Settings\All Users\Application Data\IBM\Communications Server

   If the classic Private directory location is selected, the following profile paths are used:

Operating System	User-Class Directory (Current User)[1, 2]	System-Class Directory
Windows 2000,Windows Server 2003, Windows XP, Windows Vista	C:\Program Files\IBM\ Communications Server\Private	C:\Program Files\IBM\ Communications Server\Private

Operating System	User-Class Directory (Current User)[1, 2]	System-Class Directory
[1]If the User Preference Manager (UPM) was set to a directory other than the default directory, Communications Server for Windows will utilize that directory to store the user–class files. System–class files are always stored in the Private directory. [2]For the classic Private directory locations, `C:\Program Files\IBM\Communications Server` is the directory where Communications Server is installed. **Note:** For SNA API Client installations on Windows x64 platforms, the directory path `Program Files` is replaced by `Program Files (x86)`.		

Note: For the SNA API, the directory path is `CS SNA API Client` and not `Communications Server`.

After selecting your application data location, click **Next** to continue with the installation.

2. Additional dialogs are included depending on which Communications Server for Windows product you are installing.
3. The **Ready to Install the Program** dialog opens. Click **Back** to change your previous settings, or click **Cancel** to terminate the installation process. Click **Install** to continue with installation.

The typical setup uses the C:\Program Files\IBM\Communications Server directory for program installation of the Server and the Remote Administration Client. C:\Program Files\IBM\CS SNA API Client is used for the SNA API Client.

Note: If there is not enough disk space on the C: drive, you are prompted to choose the custom installation setup type in order to choose an alternate installation destination.

After installation is complete, the **Installation Complete** dialog opens. Click **Finish** to exit the installation process.

After installation is complete, you are prompted to reboot the computer. You must reboot the computer before configuration changes take effect and you can use Communications Server for Windows product.

Note: **Typical** installs the most common features for the applicable emulators.

Custom Installation

Though the default feature selection for a custom setup is the same as for a typical setup, a custom configuration allows you to modify feature selection for your system. To continue with the custom installation setup:

1. Click the button to choose **Custom** setup type. Click **Next** to continue.
2. The **Language to be installed** dialog appears. Only one language can be installed.
3. The **Custom setup** dialog opens and asks you to select the program features that you want to install. Some features have subfeatures available. To view the subfeatures for a particular feature, click the plus sign (+) to the left of the feature name.

   Included in the Custom Setup window are **Feature Descriptions**. You can view the description of any feature by clicking on that feature and then reading the description section to the right of the feature selection tree. The feature description gives basic information about each feature, as well as the disk space

required for installation. For more detailed information on disk space requirements for each feature, click the **Disk Space** button. For a description of available features, see "Feature Selection." For a description of feature installation options, see "Feature Installation Options" on page 29.

4. From the **Custom Setup** dialog, you can change the directory where the Communications Server for Windows is installed. Click the **Change** button to choose another installation directory.

   Note: For the 64-bit operating system, the Communications Server for Windows install location for the SNA API Client cannot be changed since it is a 32-bit product.

5. After making your feature selection choices and confirming the installation directory, click **Next** to continue with the installation.

   Note: If there is not enough space on the destination drive, you are prompted to choose another location.

6. The **Application Data Location** dialog opens. Select from the following application data location options:
   - All users' common application data folder (All Users\Application Data)
   - Classic private directory

   See "Typical Installation" on page 26 for information on the profile paths used for each application data location. Click **Next** to continue.
7. Additional dialogs are included depending on which Communications Server for Windows product you install.
8. The **Ready to Install** dialog opens. Click **Install** to complete the installation.

Feature Selection

The features and subfeatures available for each of the Communications Server for Windows products are described below.

Note: In the custom setup window, if the icon to the left of the feature name is white, that feature and all of its subfeatures will be installed. If the icon appears grey, that feature or one or more of its subfeatures will not be installed.

There are no features available for the SNA API Client.

Table 4. Remote Administration Client features

Feature	Description	Subfeatures	Default
SNA Administrative and PD Aids	SNA-specific Administrative and Problem Determination Aids	• Display SNA Sense Data • SNA Node Configuration • SNA Node Configuration Verification • SNA Node Operation	Yes
Common Administrative Problem Determination Aids	Optional Administrative Tools	• Log Viewer • Information Bundler	Yes

Table 5. Communications Server features

Feature	Description	Subfeatures	Default
SNA Administrative and PD Aids	SNA-specific Administrative and Problem Determination Aids	• Display SNA Sense Data • SNA Node Configuration • SNA Node Configuration Verification • SNA Node Operation	Yes
Common Administrative Problem Determination Aids	Optional Administrative Tools	• Log Viewer • Information Bundler	Yes
IBM Global Security Kit	IBM Global Security Kit (GSKit) provides SSL and TLS connectivity	n/a	No
LLC2	IBM Communications Server provides its own IEEE 802.2 network protocol for the Local Area Network (LAN).	n/a	Yes
Documentation	Online documentation	n/a	No
Client images	Images of the SNA API Client and the Remote Administration Client used to install the client from the server.	n/a	No
SDK	Communications Server Software Developer's Kit.	n/a	No
Trace facility	Enables you to log trace information for certain server functions	n/a	Yes (required)

Feature Installation Options

Each feature and subfeature allows several installation options. To view the options available for each feature, click on the drop down icon to the left of the feature name. Select the desired installation type by clicking on it in the drop down menu. A description of each possible installation option follows:

- **This feature will be installed on local hard drive** selects this feature for installation to the local hard drive.
- **This feature, and all subfeatures, will be installed on local hard drive** selects the main feature and all of its associated subfeatures to be installed on the local hard drive.

- **This feature will be installed to run from CD** selects the feature to run from source where source medium is the CD-ROM at the local workstation. This option is only available for top-level features and installs only the base files needed to run the feature.

  Note: If you are installing from a network, this option instead displays as, **This feature will be installed to run from network**.
- **This feature, and all subfeatures, will be installed to run from the CD** selects the main feature and all associated subfeatures to run from source where source medium is CD-ROM at the local workstation. This installation option installs only the base files needed to run the features.

  Note: If you are installing from a network, this option instead displays as, **This feature, and all subfeatures, will be installed to run from the network**.
- **This feature will be installed when required** places a shortcut on the Communications Server for Windows menu allowing the feature to be installed when the shortcut is selected. This installation option is also called *advertisement*.
- **This feature will not be available** deselects the feature for installation or advertisement.

Note: If no installation choice is made, features are installed to the local hard drive.

Silent Installation

Communications Server for Windows is installed silently by passing command-line parameters through setup.exe to the MSI (Microsoft Software Installer database) package. When running a silent installation, the user does not provide input using dialogs or see a progress bar during the installation process. Instead, installation occurs automatically using either a typical configuration or a custom configuration created during initialization file processing.

Note: When migrating through silent installation, all active Communications Server for Windows sessions and actions will be closed without any prompting.

For details on initialization file processing, see "Installing Communications Server for Windows Using an Initialization (response) File." For information on performing a silent installation using setup.exe command-line parameters, see "InstallShield Professional for Window Installer (IPWI) command-line parameters" on page 39.

Installing Communications Server for Windows Using an Initialization (response) File

Communications Server for Windows provides an optional method of customization that allows property values and feature installation choices made during one installation to be automatically applied during subsequent installations. The initialization file (.ini) contains the properties and options for Microsoft Software Installer to use as initialization choices so that subsequent installations do not require users to provide installation input using dialogs. Then, future installations can be set to run silently using an initialization file.

Note: The common launchpad is not used for this type of installation. Instead, the setup.exe for each product is invoked directly.

Administrators create, save, and implement initialization files using command-line parameters. Communications Server for Windows provides the following command-line parameters:

- SAVEINI
- ONLYINI
- USEINI
- REMOVEINI

Each parameter, with a corresponding usage description, is described in the following sections.

Two sample initialization files are included on the Communications Server for Windows CD-ROM. These sample initialization files can be used during the installation if your workstation configuration matches the definitions in the sample. Communications Server for Windows includes the following sample .ini files:

- **typical.ini** installs a typical setup
- **custom.ini** installs a custom setup to a user-defined path and includes the installation of the French version

The samples are defined for a first-time installation of Communications Server for Windows. If you already have Communications Server for Windows product previously installed, the samples will not work; you will need to create a unique initialization file to match your workstation.

Note: To ensure successful initialization file processing, use all syntax examples exactly as described. The samples list all features including internal features. If you change your choice of features, it is recommended that you create your own .ini file in order to keep track of which features are sub-features of other features.

The remainder of this section provides information on the server product in Communications Server for Windows.

Parameter Descriptions

Parameters are passed through setup.exe to the MSI (Microsoft Software Installer database) package using the /v command-line parameter. Within the /v"..." argument, you can specify parameters such as SAVEINI, ONLYINI, and USEINI which all take a file path. If a path containing spaces is specified, it must be in quotes and preceded by a backslash within the /v"..." argument, as shown in the following examples. For more information on the /v command-line parameter, see "InstallShield Professional for Window Installer (IPWI) command-line parameters" on page 39.

SAVEINI

This parameter designates an initialization file to be created and installs Communications Server for Windows. To create an initialization file without installing Communications Server for Windows, see the description of the command-line parameter "ONLYINI" on page 32. The .ini file that is saved contains property values and feature installation choices that can be passed on to future installations of Communications Server for Windows. By default, the .ini file is not removed when Communications Server for Windows is uninstalled. The following command creates the .ini file and installs Communications Server for Windows:

Syntax:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\"
  SAVEINI=%temp%\cswin_server.ini\""
```

where E: is the CD-ROM drive and the .ini file is saved in the system `temp` directory. If the directory does not exist, the .ini file will not be created.

Usage Description: An administrator creates an initialization file by using the SAVEINI command-line parameter and running either a typical or custom Microsoft Software Installer installation for Communications Server for Windows (see "Installing to a Local Hard Drive" on page 25). Upon completion of the installation, the .ini file containing all of the property values and feature installation choices designated during the custom setup is saved in the directory specified in the SAVEINI parameter.

ONLYINI

This parameter designates that a .ini file be created without installing Communications Server for Windows. Upon completion of the custom setup, Microsoft Software Installer prompts you to exit the installation program without installing Communications Server for Windows.

Syntax:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\"
  SAVEINI=\"%temp%\cswin_server.ini\" ONLYINI=1"
```

where E: is the CD-ROM drive and the .ini file is saved in the system `temp` directory. If the directory does not exist, the .ini file will not be created.

When ONLYINI is set to 1, the .ini file is created without the product installation. When ONLYINI is set to 0, the parameter is inactive, and product installation occurs.

Usage Description: An administrator can use this parameter to create several unique .ini files to meet the needs of different types of users without performing an installation each time an initialization file is created.

Note: The SAVEINI and ONLYINI parameters can be used to create a .ini file for either a custom or typical installation. If using ONLYINI, installation of Communications Server for Windows is always suppressed after creating the .ini file regardless of installation choice.

USEINI

This parameter designates that a previously created .ini file be used to determine property values and feature installation choices for a current installation.

Syntax:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\"
  USEINI=\"%temp%\cswin_server.ini\""
```

where E: is the CD-ROM drive and the .ini is the initialization file you wish to designate.

Usage Description: In order to apply the initialization file you created using SAVEINI, or one of the sample initialization files included with Communications Server for Windows, you must use the USEINI command-line parameter. USEINI can be used to run either a typical or custom installation for Communications Server for Windows (see "Custom Installation" on page 27). If you choose custom setup, all property values and feature installation options designated in the specified initialization file are selected for installation.

Note: Unless installation is set to run silently, users can override values designated in the .ini file by running either a custom or typical installation setup. Any changes made during setup will alter the .ini file for future installations. See "Silent Installation Using Initialization File Processing" on page 34 for details on running a silent installation.

Though you can specify a particular initialization file to be applied with the USEINI parameter, use of initialization files is not required for remote installations. If no initialization file is specified during a remote installation, a typical installation results.

REMOVEINI

This parameter designates that the .ini file be removed when Communications Server for Windows is uninstalled from the system. If this parameter is not set or if any value other than that designated in the syntax is used, the initialization file remains on the system after Communications Server for Windows has been removed.

Syntax:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\" REMOVEINI=1"
```

where E: is the CD-ROM drive.

Usage Description: This parameter can be used at any time after you have created the .ini file. It removes the .ini file when Communications Server for Windows is uninstalled from the system. The .ini file is only removed if the occurrence of Communications Server for Windows that is being uninstalled was originally installed using the .ini file that is designated for removal.

Using System Variables and UNC Paths with Parameters

Parameters can be set to use a system variable instead of a directory path—for example, the variable `%cswin_ini%` can replace the `C:\temp\cswin_server_ini` path.

SAVEINI: In the following syntax examples, the ONLYINI parameter is set to active. The TRANSFORMS parameter launches the setup GUI in the specified language.

Using an environment variable:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\" SAVEINI=%cswin_ini%\
 MyCustom.ini ONLYINI=1 TRANSFORMS=\"1033.MST\""
```

Using UNC path names:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\"
  SAVEINI=\\MachineName\cswin_ini\MyCustom.ini ONLYINI=1
  TRANSFORMS=\"1033.MST\""
```

The shared UNC path must have write access.

USEINI: Using an environment variable:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\"
  USEINI=%cswin_ini%\MyCustom.ini ONLYINI=1 TRANSFORMS=\"1033.MST\""
```

Using UNC path names:

```
E:\Server\setup.exe /v"/L*v \"%temp%\cswin_serverinst.log\"
  USEINI=\\MachineName\cswin_ini\MyCustom.ini ONLYINI=1
  TRANSFORMS=\"1033.MST\""
```

The shared UNC path must have write access.

Silent Installation Using Initialization File Processing

In order to ensure that property values and feature installation options designated in the initialization file are not overridden by users or to enhance the ease of installation, you can apply initialization files during silent installations.

To perform a silent installation using initialization file processing, type the following command:

```
E:\Server\setup.exe /s /v"/L*v \"%temp%\cswin_serverinst.log\"
  USEINI=\"C:\Program Files\IBM\Communications Server\cswin.ini\" /qn"
```

This process passes the silent installation command-line parameter (`/qn`) through setup.exe to the MSI package. For more information about setup.exe command-line parameters, see "InstallShield Professional for Window Installer (IPWI) command-line parameters" on page 39. This parameter can also be added to commands that use system variables instead of path names.

Administrative installation

An administrative installation copies a source image of Communications Server for Windows installation files onto a network drive. The resulting location of this source image is called the installation point. After you complete an administrative installation, any user connected to the network can install Communications Server for Windows to their own workstation by pointing to the installation point and running the setup. An administrative installation offers installation directly to the user's system from the network server.

To begin an administrative installation, disable the AutoPlay function on your system or simply close the Communications Server for Windows welcome window when it opens. With the CD-ROM in the CD-ROM drive:

1. Open a command prompt and switch to the Communications Server for Windows installation directory by typing

   ```
   E:
   ```

   then

   ```
   cd Server
   ```

   at the command prompt (where E: is the CD-ROM drive).
2. From this directory, type:

```
setup.exe /a
```

3. The **Microsoft Software Installer welcome** dialog for Communications Server for Windows opens. Click **Next** to continue with the installation.
4. The **License Agreement** dialog opens. Click the button to accept the terms of agreement. You can print the license agreement by clicking **Print**. If you decline the license agreement, the installation process terminates. Click **Next** to continue.

   Note: Windows administrators will not have the option to accept the license agreement on behalf of all users.
5. The **Network Location** dialog opens. You can type the desired network installation point in the command line or click **Change** to browse for a location.
6. Click **Install** to complete the installation process.

Note: To remove the source image of Communications Server for Windows from your network server you must manually delete the source image directory from the network location.

Installing from Network Server

After the administrative installation is complete, any user connected to the network can install Communications Server for Windows from the network server. To install from the network server:

1. Click **Run...** on the Windows **Start** menu.
2. Type

   ```
   X:\MyLocation\setup.exe
   ```

   in the command line (where X: is your network server and MyLocation is the installation point designated in the administrative installation) or click **Browse** to browse for the location on the network.
3. The **Microsoft Software Installer welcome** dialog opens. Proceed with the installation as described in "Installing to a Local Hard Drive" on page 25.

Installing Patches

Communications Server for Windows will not be producing MSI fix packs but instead, will provide new CSD versions.

Maintenance installation of Communications Server for Windows

After you have successfully installed Communications Server for Windows on your system, users can perform maintenance installations to their Communications Server for Windows program. The maintenance installation utility has three functions:

- **Modify** allows users to change their feature selection options. For details on changing feature tree selections, see "Feature Selection" on page 28.
- **Repair** analyzes the current configuration of Communications Server for Windows and either repairs or reinstalls damaged features.
- **Remove** allows users to remove Communications Server for Windows from their system.

  Note: To remove a source image of Communications Server for Windows created during an administrative installation, you must manually delete the source image directory from the network drive.

When the **Program Maintenance** dialog opens, select **Modify**, **Repair**, or **Remove** and click **Next**.

Note: To successfully run maintenance installation, the Communications Server for Windows installation image must be available on either the CD-ROM or on the network server. If you installed from a network server, the installation image must still be present at the original network location. If the installation image is not present, when you use the **Modify** or **Remove** utility to add features or to remove Communications Server for Windows from your system, you may receive one of the following error messages:

- The feature you are trying to use is on a CD-ROM or other removable disk that is not available.
- The feature you are trying to use is on a network resource that is not available.

To continue with maintenance installation you must either insert the CD-ROM or browse the network to find the new location of the installation image.

Remote Installation of Communications Server for Windows

Communications Server for Windows supports remote installation using either Tivoli Software Distribution or Microsoft Systems Management Server (SMS). Remote installation and uninstallation can be performed in a normal mode (attended) or silent mode (unattended).

Considerations for Tivoli Software Distribution Support

This section provides information about creating a software package for Communications Server for Windows and customizing it, using the Software Package Editor. This information is intended for administrators and software package authors only.

With Tivoli Software Distribution, software packages can be created from Microsoft Software Installer (MSI) installation packages. This feature is part of the Tivoli Software Package Editor. MSI packages or MSI patches can be imported into the Software Package Editor, and can be edited, reconfigured, and built as Tivoli software packages. The MSI support in the Software Package Editor simplifies the configuration steps that were necessary in previous versions of Tivoli.

The Communications Server for Windows installation image is distributed from a central repository to the target's temporary image repository (for example, C:\Temp). The product is then installed from the temporary repository. Generally, the images are not retained at a target after the successful distribution because of disk space considerations. However, you can configure to save the images at the target.

Creating the Installation Object

Use the following procedure to create the Communications Server for Windows MSI installation object, using the Tivoli Software Package Editor.

1. From the Software Package Editor menu, click **Tools → Importer → Install MSI Product**.
2. The first screen shows general information about the MSI import procedure. Click **Next**.

3. Enter the location and the name of the MSI Installation package. Use the IBM Communications Server for Windows installation CD location or copy the contents of the CD into a local folder.

   The Importer tool then reads the information from the MSI package. The settings of the package can be changed after this operation.
4. The product name, version, and manufacturer fields are shown in the next panel. Click **Next** to proceed.

   Note: The installation destination path on the target workstation (where Communications Server for Windows is to be installed) cannot be selected in this wizard panel. This information must be set in the package properties (see "MSI Installation Package Properties").
5. In the next panel, you can specify the target image path, which is the location where the installation images are to be copied on the target workstation. For example, you can set the target path as C:\Temp\CSWIN. You must also specify the source image path, which defines the current location of the installation image.

   Notes:

   a. If you select the **Redirected Installation** option, you should ensure that the target image path can be accessed by the target workstation at the time of installation.

   b. If you select the **Keep Images** option, the installation image will not be deleted from the target workstation after the installation.

   c. If you select the **All Users** option, the installed product will be available for use by all users on the workstation. If it is not selected, only the user who is logged on at the installation time can use the product.

   Click **Next** to proceed.
6. The next panel shows the features of the MSI image in an expandable tree format. Changes to the image can be made now or at a later time. Click **Finish** to create the Communications Server for Windows MSI installation object.

MSI Installation Package Properties

Once the MSI installation package is added to the installation objects, you can set the properties of the package. In the installation objects list, right-click on the MSI package that you added to display the properties panel.

You might need to customize the following items in a Communications Server for Windows software package:

- Features to be installed.
- The product installation directory.
- The user's application data directory, for user-class files: **Classic Private**, **User's Application Data**, or **All Users**. See "Typical Installation" on page 26 for information about default system and user data directories for the supported Windows operating systems.
- The language to be installed.
- Administration and configuration ID
- Number of client access licenses
- Configuration option.

Features: To select the features to be installed, click the **Features** tab. Right-click on the leaf of the expandable feature tree. You can then select or deselect features from those displayed.

Product Installation Location: To set the Communications Server for Windows installation directory, click the **Properties** tab. The install location is defined by the property INSTALLDIR, which must be defined in order to have a successful installation. For example, you can set the property as follows:

INSTALLDIR="C:\Program Files\IBM\Communications Server"

In addition, software package variables like $ (system drive) can be used to define the **INSTALLDIR** property.

There is no default value for this property.

User's Application Data Directory: On the Properties tab, you can set the default directory in which to save users' application data, such as workstation profiles, keyboard map files, and macros. This location is defined by the **APPDATALOCATION** property. The following example sets the Communications Server for Windows classic ...\private directory:

APPDATALOCATION="2"

The APPDATALOCATION can take the following values:

- **1**

  Files are stored in the All Users application data directory.
- **2**

  Files are stored in the classic ...\private directory.

Installing Languages: By default, Communications Server for Windows is installed with English language support. You can alternatively install the languages listed in Table 7 on page 42 from the Properties tab. For example, Set **SELECTED_LANG="1031"** for German.

Administration and configuration ID: On the **Properties** tab, you can set the user ID you wish to add to the administration and configuration group IBMCSADMIN (for server only; not Remote Administration Client and SNA API Client).

ADMINID="<*userdomain*>"\"<*userName*>"

Number of client access licenses: On the **Properties** tab, you can set the number of licenses (for sever only, not Remote Administration Client and SNA API Client).

LICENSES_PURCHASED="<*number*>"

Configuration options: On the **Properties** tab, you can set the configuration option to be used (for the SNA API Client only; not Remote Administration Client or the Server). The following example sets it to the Local .INI file:

CONFIG_OPTION="3"

where

1 Directory Services

2 Lightweight Directory Access Protocol

3 Local .INI file [INSTALLDIR]CSNTAPI.INI

When you have completed the Features and Properties modifications, click **OK** in the Properties panel and save the software package (as a .SP or .SPD file). The software package can then be imported into a Tivoli profile and distributed to the endpoints.

Remote Installation Using SMS

A remote installation using SMS consists of the following steps:

1. Perform an administrative installation to copy Communications Server for Windows installation files to the network (see "Administrative installation" on page 34).
2. Create an SMS package containing the Communications Server for Windows installation software.
3. Create an SMS job to distribute and install the software package.

For detailed and up-to-date instructions on installing and deploying Communications Server for Windows using SMS, refer to the SMS product documentation provided at http://www.microsoft.com/smsmgmt.

Remote Installation Using Common Launchpad

If you plan to run the Common Launchpad executable from a remote location, the location must be mapped to the network drive letter.

InstallShield Professional for Window Installer (IPWI) command-line parameters

InstallShield Professional uses setup.exe as the bootstrap loader to call the Microsoft Software Installer. Setup.exe can accept command-line parameters that allow you to perform administrative installations, run silent installations, and complete other administrative tasks. Using the /v parameter, other parameters can also be passed through setup.exe to the Microsoft Software Installer database (MSI package). For information on using the /v parameter to perform initialization file processing, see "Installing Communications Server for Windows Using an Initialization (response) File" on page 30.

By default, setup.exe creates a verbose installation log with the file name cswininst.log, and places it in the folder named by the environment variable `%temp%`. This behavior is overridden when command-line arguments are passed to the Windows Installer using the /v parameter, as described in "Parameter Descriptions" on page 40.

Note: If `%temp%` points to a nonexistent folder and the /v flag is not used to override the default parameters passed to the Windows Installer, then setup.exe will fail.

Setup.exe accepts the command-line parameters listed in Table 6. Descriptions of each parameter are listed in "Parameter Descriptions" on page 40.

Table 6. InstallShield Command-Line Parameters

Parameter	Description
/v	Passes parameters to MSI package.
/s	Causes setup.exe to be silent.
/l	Specifies the setup language.

Table 6. InstallShield Command-Line Parameters (continued)

Parameter	Description
/a	Performs administrative installation.
/j	Installs in advertise mode.
/x	Performs setup uninstall.
/f	Launches setup in repair mode.
/w	Setup.exe waits for the installation to finish before exiting.
/qn	A Windows Installer MSI parameter that causes everything but setup.exe to be silent. This sets the user interface level to zero.

Parameter Descriptions

Passing parameters to the MSI package

/v

The /v command-line parameter enables you to pass parameters supported by Microsoft Software Installer through setup.exe to the MSI package. For example, you can create and save a verbose log file to a location of your choice by passing the /L parameter through setup.exe to the MSI package. To create the log file, type:

```
E:\Server\setup.exe /v"/L*v\"%temp%\cswininst.log\"
```

where E: is your CD-ROM drive. For more information on supported command-line parameters and specific usage examples refer to the Web site http://www.msdn.microsoft.com.

Note: The /v argument must be the last InstallShield parameter on the command line. Though supported Microsoft Software Installer parameters may be passed through to the MSI package, no InstallShield command-line parameters can follow the /v argument.

Running setup.exe silently

/s

To prevent setup.exe from displaying a progress bar, use the /s command-line parameter. To have setup run silently with no dialogs, pass the Microsoft Software Installer /qn command-line parameter through setup.exe using the /v parameter. Refer to Microsoft's documentation of command-line parameters for other /q user interface options. To run a silent installation, type:

```
E:\Server\setup.exe /s /v"/L*v \"%temp%\cswininst.log\" /qn"
```

where E: is your CD-ROM drive. This example performs a typical installation of Communications Server for Windows server to your local hard drive without showing a progress bar or input dialogs. For information on property values and features included in a typical installation, see "Typical Installation" on page 26 and "Feature Selection" on page 28.

Note: You can pass an initialization file to the MSI package and run the installation silently using the /s /v /qn parameters in the following command:

```
E:\Server\setup.exe /s /v" /L*v \"%temp%\cswininst.log\"
USEINI=\"C:\Communications Server\cswin.ini\" /qn"
```

For more information on initialization file processing see "Installing Communications Server for Windows Using an Initialization (response) File" on page 30.

To install silently from source, where source medium is a network server, use the /s /v /qn parameters after pointing to the installation point in the command line. See "Administrative installation" on page 34 for more information on performing administrative installations and installing from a network server.

To uninstall Communications Server for Windows server silently, use the /s parameter in conjunction with the /x parameter as shown in the following example:

```
E:\Server\setup.exe /s /x
```

where E: is the location of the Communications Server for Windows installation directory. For example, the code from the CD-ROM drive.

Specifying the setup language

/l

The /l command-line parameter enables you to specify what language to use during setup by using the appropriate decimal language identifier. For a list of language identifiers, see "National Language Support (NLS) Abbreviations and Language Codes" on page 42. For example, to change the setup language to Spanish, type:

```
E:\Server\setup.exe /l"1034"
```

Administrative installation

/a

Administrative installation installs a source image to the network server. This enables users with access to the network to install Communications Server for Windows directly from the network server.

Advertise mode

/j

Advertisement enables users to install features of Communications Server for Windows when they need them rather than during setup. Features that are available for installation are advertised with shortcuts on the user's system for later installation.

Note: Communications Server for Windows does not support Advertise mode.

Uninstall mode

/x

Uninstall mode removes Communications Server for Windows from your system.

Repair mode

/f

Launching setup.exe in the repair mode checks the key file of every installed feature and reinstalls any feature that is determined to be missing, corrupt, or an older version.

National Language Support (NLS) Abbreviations and Language Codes

Table 7 lists the abbreviations and language codes that are shipped with Communications Server for Windows. These codes are used in conjunction with the /l command-line parameter and in language string tables and dialogs. For more information on the /l command-line parameter see "InstallShield Professional for Window Installer (IPWI) command-line parameters" on page 39.

For more information on using transform files, refer to http://www.msdn.microsoft.com.

Table 7. National Language Support Abbreviations and Language Codes

Language	Server language abbreviations	Language Codes (shipped as *.mst transform files)
English (United States)	enu	1033
French (France)	fra	1036
German (Germany)	deu	1031
Italian (Italy)	ita	1040
Japanese (Japan)	jpn	1041
Portuguese (Brazil)	ptb	1046
Spanish (Traditional-Spain)	esp	1034
Korean (Korea)	kor	1042
Simplified Chinese	chs	2052
Traditional Chinese	cht	1028

Chapter 4. Configuring Communications Server for Windows

This chapter explains the basics of how to use the configuration program provided with Communications Server for Windows. You can use the GUI interface or you can modify the fn.acg configuration file using the ASCII editor.

SNA Node Configuration provides configuration options that enable you to define SNA nodes and associated resources. You can define the following SNA characteristics:

- Network functions such as APPN, host connectivity, or LU 0
- Link characteristics
- Session partners and parameters
- Transaction programs on the local node that can be accessed through the SNA network (also known as target TPs)

The set of objects that must be configured depends on the environment in which Communications Server operates and the specific functions Communications Server is to support.

Back up your configuration files periodically in case a problem occurs with the configuration. By default, your configuration files (.ACG files) are located in the **\private** subdirectory of your Communications Server installation directory. You can also save them in documents and settings.

You can get to this information from a configuration scenario by selecting any step and pressing F1. For additional information, refer to the configuration examples in SampleConfigurations of your Communications Server installation directory.

Deciding What to Configure

How you configure and use Communications Server depends on the task you are trying to accomplish. Scan the list of common configuration types below to find the section of this chapter that discusses your configuration needs. Each configuration scenario is self-contained and divided into steps.

Configuration Type	Refer To:
Use Communications Server as a SNA gateway allowing clients to access host applications through this server	"SNA Gateway Configuration" on page 45
Use Communications Server as a TN3270E server	"TN3270E Server Configuration" on page 46
Use Communications Server as a TN5250 server	"TN5250 Server Configuration" on page 47
Define an APPN network node (with HPR and connection networks)	"APPN Network Node Configuration" on page 48
Define the node as a dependent LU requester (DLUR) over an APPN network, or as a gateway between a downstream DLUR and an upstream DLUS (dependent LU server)	"DLUR/DLUS Configuration" on page 49
Define an AnyNet SNA over TCP/IP gateway	"AnyNet SNA over TCP/IP Gateway Configuration" on page 50

© Copyright IBM Corp. 1997, 2007

Configuration Type	Refer To:
Configure Communications Server to support SNA API Clients running APPC applications	"SNA API Client Configuration for APPC" on page 51
Configure Communications Server to support SNA API Clients running 3270 or other LUA applications	"SNA API Client Configuration for LUA" on page 56
Run APPC or CPI-C applications or 5250 emulation to an iSeries system	"CPI-C or APPC Configuration" on page 60
Access host applications using a 3270 emulator	"3270/LUA applications" on page 61
Define a focal point	"Focal Point Configuration" on page 62

As a starting point, choose the scenario that most closely matches your configuration. The online *Network Administration Guide* contains information that can help you decide which type of configuration you should use. You can then use the information in the online help to alter the configuration as necessary.

You can also use these scenarios to create a basic configuration for multiple machines. Refer to the online *Configuration File Reference* for detailed information about customizing your configuration.

Getting Started

To begin configuring, follow these steps:

1. Click **SNA Node Operations** in the Communications Server folder. On the toolbar, click **Configuration Utility**.
2. Select **New** on the Welcome to Communications Server Configuration! window to create a new configuration. Click **Next**.
3. Select the scenario you want from the Choose a Configuration Scenario window or click **Advance** to have all configuration options available. Click **Finish**.
4. A window appears that lists the steps you need to follow and shows a diagram of the resources you need to configure. Click the help button for a step to get detailed instructions. Click the action button for a step to bring up the configuration window and add a definition for that step.
5. To modify a definition, double-click its entry in the tree-view diagram.

Testing Your Configuration

The steps you need to take to test your configuration are listed in each scenario and described fully online. Usually, testing involves the following steps:

1. Start the node

   Click **Start/Stop Node** on the SNA Node Operations toolbar to start the local node. Specify a configuration with which to start the node. Select the file you have just configured and click **Open**. It may take a moment for the node to start and to establish all the defined links.

2. Verify that the link is active

   Once the node has started, you should see a view of the active node with the control point name displayed that you specified in Step 1. (It may take a few moments to establish the link.) To verify that the link you configured is active:

a. Find the **Host Resources** icon in the left side of the SNA Node Operations window. Expand the list of resources by clicking the plus mark (+) next to the icon.

b. Click **Connections** (for CPI-C and APPC configurations, click **Peer Connections**. Defined links are displayed in the right side of the SNA Node Operations window.

c. To display details about the displayed links, click the **Details** icon on the toolbar.

3. Establish a session

   For most gateway configurations, have clients establish a session through the gateway to the host. The client configurations need to specify the gateway's network addressing information, such as the token-ring address for the gateway's token-ring card.

   For APPN network node, AnyNet SNA over TCP/IP, and CPI-C and APPC configurations, you can use the Check Connection (APING) application from the Communications Server program group on the desktop to verify that a remote LU in the network is reachable. When you invoke the application, specify the fully qualified partner LU. Click **OK** to continue. The application will send test packets of data to the remote node's APINGD application. These packets will in turn be returned to the local application.

SNA Gateway Configuration

This section describes the steps to configure Communications Server as an SNA gateway for passing sessions between SNA clients and hosts.

By configuring an SNA gateway, you are able to exchange data between downstream client workstations and a Systems Network Architecture (SNA) mainframe host computer. The gateway performs the function of a concentrator, which makes LU resources from one or more hosts and host PUs available to downstream SNA client workstations over any combination of communications media supported by Communications Server.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields. If you are using DLUR/DLUS, refer to "DLUR/DLUS Configuration" on page 49.

Before You Begin

Before you begin this procedure, collect the following information:

- A local node ID (XID) corresponding to the host's PU (for switched physical media) *or* a local secondary station address corresponding to the host's PU (for non-switched physical media)

A local node ID consists of a block ID followed by a PU ID.

- At least one local LU address available on the same host PU
- Addressing information for contacting the host's communication controller (such as the token-ring, ethernet or TCP/IP address)

- The physical network type your downstream clients are using (for example, ethernet)
- If you are using AnyNet SNA over TCP/IP:
  - The IP address of your machine
  - The IP address of your gateway or partner
  - The domain name suffix administered by your site administrator.

  See the online *Network Administration Guide* for more information.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

If you have wizards enabled, a wizard will appear for this scenario instead of the window described in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Device configuration
3. Define host connections and LUs
4. Define gateway clients
5. Start the node
6. Verify that the link is active
7. Establish a session

TN3270E Server Configuration

By configuring a TN3270E server, you are able to exchange data between downstream TCP/IP client workstations running TN3270E (or TN3270) applications and a Systems Network Architecture (SNA) mainframe host computer. The server performs the function of a concentrator, making LU resources from one or more hosts and host PUs available to downstream TN3270E (or TN3270) sessions over any combination of communications media supported by Communications Server.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields. If you are using DLUR/DLUS, refer to "DLUR/DLUS Configuration" on page 49.

Before You Begin

Before you begin this procedure, collect the following information:

- A local node ID (XID) corresponding to the host's PU (for switched physical media) *or* a local secondary station address corresponding to the host's PU (for nonswitched physical media)

A local node ID consists of a block ID followed by a PU ID.

- At least one local LU address available on the same host PU

- Addressing information for contacting the host's communication controller (such as the ethernet or TCP/IP address)

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

If you have wizards enabled, a wizard will appear for this scenario instead of the window described in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Device configuration
3. Define host connections and LUs
4. Define TN3270E server properties
5. Start the node
6. Verify that the link is active
7. Establish a session

TN5250 Server Configuration

By configuring a TN5250 server, you are able to exchange data between downstream client workstations on a TCP/IP network running TN5250 applications and an iSeries on an SNA network. The server performs the function of a gateway, making iSeries display sessions available to downstream TN5250 clients over any combination of communications media supported by Communications Server.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin" on page 46. If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin this procedure, collect the following information:

- At least one iSeries fully-qualified control point name
- If the iSeries requires it, a user ID and password needed to access that iSeries
- A partner LU for LEN connections (optional)
- Addressing information for contacting the iSeries (such as the ethernet or TCP/IP address)

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Device configuration
3. Peer Connection configuration
4. iSeries server definition

5. TN5250 defaults configuration
6. Start the node
7. Verify that the link is active
8. Establish a session

APPN Network Node Configuration

This section describes how to configure the local node as an APPN network node. Network nodes are responsible for maintaining and routing information through an APPN network. They maintain a directory of all resources (LUs) available on directly attached end nodes, as well as maintain a topology of all network nodes in the APPN network. When an independent LU 6.2 session is requested by a node in the network, the network node is responsible for locating the remote resource requested by the session and, once located, for establishing and routing the session between the two nodes.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin this procedure, collect the following information:

- The physical communications media (for example, ethernet, TCP/IP or X.25) over which links to the APPN network will be established.
- If any low-entry networking (LEN) nodes will be served by this network node, you must obtain each LEN node's control point (CP) name, as well as any LU names on that node which should be made available to other nodes in the network. LEN nodes do not support APPN, so the network node must ensure that the LEN node's resources are reachable.
- Determine if this network node will be establishing a link with another node or if other nodes will be calling into this network node. If this network node will establish links into other nodes, the addressing information for those nodes is required (for example, ethernet adapter or TCP/IP address).

You can use the discovery function (available over LAN devices only) to do this.

- Determine if this network node will participate in a connection network (available over LAN, EEDLC IPv4 or EEDLC IPv6 devices only). However, protocols cannot be mixed in the same connection network (EE and LAN cannot be in the same connection network). If so, the connection network name is required. Connection networks allow non-adjacent nodes to establish direct links to one another over which sessions will be routed. This method is faster than having the session routed over multiple nodes in the APPN network.
- If you are using AnyNet SNA over TCP/IP:
  - The IP address of your machine
  - The IP address of your gateway or partner
  - The domain name suffix administered by your site administrator.

    See *Network Administration Guide* for more information.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node configuration
2. Device configuration
3. Connection configuration
4. Start the node
5. Verify that the link is active
6. Establish a session

DLUR/DLUS Configuration

This task describes the steps to configure Communications Server to support dependent LU sessions over APPN networks. This feature requires that the Communications Server act as a dependent LU requester (DLUR) that requests routing services from a dependent LU server (DLUS), which resides on a host.

Communications Server supports DLUR for local sessions and devices, as well as downstream sessions and devices. Because of these two types, this task includes two subtasks:

- Local DLUR Support
- DLUR Support for Downstream LUs

In either case, the local node must be configured to connect into an APPN network; to do this, use the APPN Network Node Configuration task (see "APPN Network Node Configuration" on page 48). You may also use this task to configure the local node as an APPN end node; to do this, on the Node Setup step, select **End node** as the node type, and disregard the Partner LU 6.2 Configuration step.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin this procedure, collect the following information:

- The fully qualified DLUS name (network name and LU name) representing your primary dependent LU server as defined in VTAM®
- The fully qualified backup DLUS name representing your backup dependent LU server, if you have one
- For local DLUR support:
  - A local node ID (XID) corresponding to the host's PU that will be served by the DLUS

A local node ID consists of a block ID followed by a PU ID.

  - At least one available LU address on the host's PU for each session

If you are using AnyNet SNA over TCP/IP:

- The IP address of your machine
- The IP address of your gateway or partner
- The domain name suffix administered by your site administrator.

  See *Network Administration Guide* for more information.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

Subtask: DLUR Support for Local LUs

1. APPN configuration
   a. Node setup
   b. Device configuration
   c. Connection configuration
2. DLUR PUs configuration
3. Start the node
4. Verify that the link is active
5. Verify that DLUR PU is active
6. Establish a session

Subtask: DLUR Support for Downstream LUs

1. APPN configuration
   a. Node setup
   b. Device configuration
   c. Connection configuration
2. Assign DLUS to clients
3. Start the node
4. Verify that the link is active
5. Verify that DLUR PU is active
6. Establish a session

AnyNet SNA over TCP/IP Gateway Configuration

This section describes the steps to configure Communications Server as an AnyNet SNA over TCP/IP Gateway. The function of the SNA over TCP/IP gateway is to route SNA traffic (independent LU 6.2) between TCP/IP and SNA networks. The SNA over TCP/IP gateway, for example, allows APPC or CPI-C applications running on another AnyNet SNA over TCP/IP access node to communicate with SNA peer nodes on an SNA network.

The AnyNet SNA over TCP/IP Gateway must be configured as an APPN network node. This will allow APPC sessions to be routed to appropriate SNA peer nodes.

If you want to route dependent LU sessions from downstream AnyNet SNA over TCP/IP workstations to upstream hosts, use the SNA Gateway function along with AnyNet SNA over TCP/IP. See "SNA Gateway Configuration" on page 45 to set up the local node for this function.

Do not proceed with Step 1 until you can answer all questions and locate **all** the information in "Before You Begin."

Before You Begin

Before you begin this procedure, collect the following information:

- The fully qualified LU name(s) (NetworkID.LUName) of the LUs (on the SNA and IP networks) that will be communicating through the AnyNet SNA over TCP/IP gateway(s).
- The IP addresses of the gateway(s) and access node(s) that will be using AnyNet SNA over TCP/IP for communication.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Configure AnyNet SNA/IP gateway device
3. Configure other devices
4. Connection configuration
5. Partner LU 6.2 configuration
6. Map the LU names to TCP/IP addresses and configure them in the **hosts** file. The **hosts** file is located in the Windows/system32/drivers/etc directory.
7. Start the node
8. Verify that the link is active
9. Establish a session

SNA API Client Configuration for APPC

This section describes the steps to configure the Communications Server for supporting SNA API Clients. SNA API Clients are separate machines from the Communications Server which have the ability to run SNA applications while using the SNA code on the Communications Server node to actually manage the session. Communications Server simply establishes and manages a session as if the origin were the local node, but forwards the application data to the SNA API Client instead of to its own application APIs.

SNA API Clients give you the ability to run SNA applications without having to install an SNA communications stack, such as the Communications Server for Windows on the same machine. Smaller, less powerful machines can thus be used to run the SNA applications while a centralized, more powerful machine can be dedicated as the SNA server for these SNA API Clients.

SNA API Clients support two types of applications: APPC (independent LU 6.2) applications and LUA API applications, such as 3270 emulators using SLI or RUI interfaces. For support of LUA applications, see "SNA API Client Configuration for LUA" on page 56. Configuring for SNA API Client support involves two sub-tasks: configuring the Communications Server to support SNA API Clients and their applications, and configuring the SNA API Client to communicate with Communications Server.

Configuring Communications Server for SNA API Client Support

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

This section describes using the configuration interface to configure SNA API Clients. If you will be using Lightweight Directory Access Protocol (LDAP) to provide configuration access across your clients, see Appendix A, "Directory Exploitation," on page 85 for an overview of LDAP as it refers to Communications Server SNA API Clients.

Before You Begin

Before you begin configuring Communications Server, collect the following information:

- The user IDs with which the API clients will connect into the local Windows system to access Communications Server.
- Determine what type of applications the SNA API Clients will run: APPC or 3270 (LUA) applications. This will help you decide which path to take when configuring Communications Server.

Communications Server may be configured to support SNA API Clients running both APPC and 3270 applications. Some of the steps will be similar between the two sub-tasks; you may be able to reuse portions of one configuration in the other.

If you are configuring Communications Server to allow SNA API Clients to run APPC applications, you must know:

- A control point name for this node. The name must be unique within the network.

If you do not know what value the control point name should take, a good rule of thumb is to use the first 8 characters of the local node's TCP/IP host name, as an example of a unique name, which is found by entering **hostname** on the command line.

- The name of the network in which this node will operate.

If you are not using APPN and do not know what the value should be, you can make up a fictitious name.

- Will this node operate in an APPN network? If so, will it function as an APPN end node or as an APPN network node? This information is used to set the control point type in Step 1.
- If you are not connecting Communications Server to an APPN network, and you are establishing a session from the local node to another node in the network (instead of receiving a session request from another node), you need the control point name of the remote node and the LU name on that node to establish your session. Step 5 makes use of this information.

- Will you establish a link from this node to another node, or will another node call into this node? If you need to establish a link with another node, you must know the addressing information for establishing that link. Use Step 3 to configure that link.
- The local LU 6.2 LUs to be accessed by clients
- The Transaction Programs (TPs), along with their attributes, to be configured on the clients

Configuring Communications Server to Support SNA API Clients Running APPC Applications

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

Configuring Communications Server to support SNA API Clients running APPC applications:

1. Node setup
2. Device configuration
3. Connection configuration
4. Local LU 6.2 configuration
5. Partner LU 6.2 configuration
6. Transaction program configuration

Configuring the SNA API Client to run APPC applications:

1. Configure global data
2. Configure server location
3. Configure transaction program definition
4. Configure CPI-C side information definition

Testing the configuration:

1. Start the Communications Server node
2. Verify that all configured links are active
3. Establish the connection from the SNA API Client

Administering User IDs

You can restrict which SNA API Clients can connect with the Communications Server by adding user IDs and global group IDs to the local group IBMCSAPI. Only clients using a user ID under this group will be allowed access to Communications Server.

To define the IBMCSAPI user group and all user IDs and passwords:

1. Invoke the Windows User Manager application to view the local system security database. From the Start menu, select **Programs > Administrative Tools**.
2. From the groups list, double-click on the IBMCSAPI group, which is created for you during Communications Server installation, to invoke the change dialog for the group. Add the user IDs and global group IDs that you want to give access to, if they are not listed.

3. Exit the User Manager application.

Configuring the SNA API Client to Run APPC Applications

Because SNA API Client configuration differs slightly between operating systems, this section will address the steps necessary to configure the SNA API Client as generally as possible so as to apply to all operating system clients.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin configuring the SNA API Client, collect the following information:

- The user IDs configured in the IBMCSAPI group on the Windows system where Communications Server resides.
- The TCP/IP address or host name of the Windows system where Communications Server resides.
- If transaction programs on the SNA API Client will need to be run by remote SNA nodes, create a transaction program definition in Step 3 under the APPC configuration subtask below.
- If you will be running an APPC application locally, what does it require in terms of session information? Does it require that you specify a partner LU name and mode name as parameters? (You can assign default local LUs and default Partner LUs. Refer to the online *Network Administration Guide* for more information) Does it require that you specify the remote transaction program to execute? Or does it require that you specify a CPI-C Side Information definition for providing all of this information? If a CPI-C Side Information definition is required, Step 4 under the APPC configuration subtask is also required.

Step 1: Configure Global Data

You can configure the user ID and password that will give the SNA API Client access to Communications Server. Choose one of the user IDs configured under the IBMCSAPI user group on the Windows system where Communications Server resides. If you do not configure the user ID and password, Communications Server prompts the user to supply it each time the client attempts to connect to the server, unless the user logs on to the Windows domain with a user ID or password from the IBMCSAPI user group.

Step 2: Configure Server Location

For TCP/IP, specify either the host name or address of the Communications Server machine. This will direct the SNA API Client to that machine when requesting services from Communications Server.

Step 3: Configure Transaction Program Definition

If an APPC or CPI-C transaction program is to be run locally by remote SNA nodes, a transaction program definition is required. In this definition, specify the TP name by which the application will be known to the SNA network. Also specify the location of the executable file within the SNA API Client's file system.

Step 4: Configure CPI-C Side Information Definition

If a CPI-C transaction program is to be run locally by a user and it asks that a CPI-C side information definition be supplied in order to provide session characteristics, then this definition is required. In the CPI-C Side Information definition, you must provide the information which the application cannot gather by itself. You must specify a symbolic destination name by which this definition will be known, and an asterisk (*) for the local LU name. This tells the Communications Server to which the SNA API Client will connect to choose on behalf of the client an LU 6.2 LU configured for SNA API Client traffic.

You should define a local LU name on the server on the SNA API Client if you want the server to run applications to the client.

You must specify the fully qualified (NetworkID.LUName) partner LU name (or partner LU alias) with which the SNA API Client will establish a session. You must also provide the remote transaction program to invoke and what mode name to use for providing session parameters.

If you specify a mode other than a provided default mode, that mode definition must exist on Communications Server as well.

When you are finished, save your configuration. The current working configuration is normally called CSNTAPI.INI, but you can save your configuration under a different name. To use a renamed configuration, set the environment variable CSNTAPI. For example, to use the configuration named *MYCONFIG.INI*, use the command `SET CSNTAPI=MYCONFIG`.

You can also use configuration files in other directories, including shared files in remote directories. To use the configuration *CSNTUSER.INI* in a remote directory *P:\MYUSER*, use the command `SET CSNTAPI=P:\MYUSER\CSNTUSER.INI`.

Testing the Configuration

Step 1: Start the Communications Server node

Start the SNA Node Operations application under the Communications Server program group if it is not already started. Click **Start** on the toolbar to start the local node. Specify the configuration file you used earlier in this task. It may take a moment for the node to start and to establish all the defined links.

Step 2: Verify that all configured links are active

If you configured any connection definitions for Communications Server, such as a host link, ensure that these start successfully by clicking on **Connections** from the toolbar. Any configured links should be shown in the list box. If they do not become active after a short period of time, ensure that the remote nodes are ready to receive link activation requests from Communications Server.

If you are configured as a network node, it may be that a remote node must start a link with Communications Server in order to connect into an APPN network, in which case no link definitions are necessary on Communications Server. Ensure that any remote nodes with links defined to Communications Server have been started.

Once Communications Server has been started and all links into the SNA network have been established, you are ready to have your SNA API Client connect with Communications Server.

Step 3: Establish the connection from the SNA API Client

If you are running an APPC or LUA application locally, a connection to Communications Server will be established when the application is started.

If you are using the SNA API Client for Windows 2000 or a later release, you can use the LU 6.2 Connection Tester utility to test the connection. Click the **LU62 Connection Tester** icon in the SNA API Client folder to start the utility. The connection tester attempts to open a connection between the client and the server using the current configuration. If the connection completes successfully, a window appears with information about the connection. If the connection does not complete successfully, the connection tester provides an error message that includes probable reasons the connection failed. The connection tester is designed to enable you to check the connection between the client and the server that LU 6.2 applications residing on the client will use. It does not determine the status of the complete connection between the client APPC node and eventual partner APPC nodes.

If you have applications on the SNA API Client to which you want to give other SNA nodes access, start the Attach Manager application. Attach Manager automatically starts transaction programs configured to start with Attach Manager.

SNA API Client Configuration for LUA

This section describes the steps to configure the Communications Server for supporting SNA API Clients. SNA API Clients are separate machines from the Communications Server which have the ability to run SNA applications while using the SNA code on the Communications Server node to actually manage the session. Communications Server simply establishes and manages a session as if the origin were the local node, but forwards the application data to the SNA API Client instead of to its own application APIs.

SNA API Clients give you the ability to run SNA applications without having to install an SNA communications stack, such as the Communications Server for Windows on the same machine. Smaller, less powerful machines can thus be used to run the SNA applications while a centralized, more powerful machine can be dedicated as the SNA server for these SNA API Clients.

SNA API Clients support two types of applications: APPC (independent LU 6.2) applications and LUA API applications, such as 3270 emulators. For support of APPC applications, see “Configuring the SNA API Client to Run APPC Applications” on page 54. Configuring for SNA API Client support involves two sub-tasks: configuring the Communications Server to support SNA API Clients and their applications, and configuring the SNA API Client to communicate with Communications Server.

Before proceeding with this configuration, locate **all** the information listed in “Before You Begin” on page 57. If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Configuring Communications Server for SNA API Client Support

Before You Begin

Before you begin configuring Communications Server, collect the following information:

- The user IDs and passwords with which the API clients will connect into the local Windows system to access Communications Server.
- Determine what type of applications the SNA API Clients will run: APPC or 3270 (LUA) applications. This will help you decide which path to take when configuring Communications Server.

Communications Server may be configured to support SNA API Clients running both APPC and 3270 applications. Some of the steps will be similar between the two sub-tasks; you may be able to reuse portions of one configuration in the other.

If you are configuring Communications Server to allow SNA API Clients to run 3270 (LUA) applications, you must know:

- A local node ID (XID) corresponding to the host's PU (for switched physical media) *or* a local secondary station address corresponding to the host's PU (for nonswitched physical media)

A local node ID consists of a block ID followed by a PU ID.

- At least one local LU address available on the same host PU
- Addressing information for contacting the host's communication controller (such as the ethernet or TCP/IP address)
- The LU pool that will be the default pool for client usage, if any

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

Configuring Communications Server to support SNA API Clients running 3270 or other LUA applications:

1. Node setup
2. Device configuration
3. Connection configuration
4. Define host connections and LUs
5. Enable SNA API Client support

Configuring the SNA API Client to run 3270 or other LUA applications:

1. Configure global data
2. Create an LUA definition

Testing the configuration:

1. Start the Communications Server node
2. Verify that all configured links are active
3. Establish the connection from the SNA API Client

Configuring Communications Server to Support SNA API Clients Running 3270 or Other LUA Applications

Administering User IDs

You can restrict which SNA API Clients can connect with the Communications Server by defining user IDs and passwords under the user group IBMCSAPI. Only clients using a user ID under this group are allowed access to Communications Server. See "Administering User IDs" on page 53 for information about how to define this group.

Configuring the SNA API Client to Run 3270 or Other LUA Applications

Because SNA API Client configuration differs slightly between operating systems, this section will address the steps necessary to configure the SNA API Client as generally as possible so as to apply to all operating system clients.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin configuring the SNA API Client, collect the following information:

- The user IDs and passwords configured in the IBMCSAPI group on the Windows system where Communications Server resides.
- The TCP/IP address or host name of the Windows system where Communications Server resides.
- The LU name of an LU 0 to 3 LU configured on Communications Server if you require a specific LU address. If you do not, you may configure the SNA API Client to simply request the next available LU in the default pool.

Step 1: Configure Global Data

You can configure the user ID and password that will give the SNA API Client access to Communications Server. Choose one of the user IDs configured under the IBMCSAPI user group on the Windows system where Communications Server resides. If you do not configure the user ID and password, Communications Server prompts the user to supply it each time the client attempts to connect to the server, unless the user logs on to the Windows domain with a user ID and password from the IBMCSAPI user group. This can cause users to be unable to receive all the output of the initial screen when connecting to the system.

Step 2: Create an LUA Definition

When creating the LUA definition, give a meaningful session name by which this definition will be known. You must configure a line within this definition for each application you want to run concurrently on the SNA API Client.

For TCP/IP, specify the host name or address of the Communications Server machine on each line.

Also, specify whether the session should use a dedicated LU or an LU from the default pool. If you specify that a dedicated LU be used, provide the LU name as configured in Communications Server. The easiest configuration is to specify that an available LU be chosen from the default pool, which was configured as the PUBLIC pool on Communications Server. If the definition requires you to specify a pool name, use PUBLIC.

When you are finished, save your configuration. The current working configuration is usually called CSNTAPI.INI, but you can save your configuration under a different name. To use a renamed configuration, set the environment variable CSNTAPI. For example, to use the configuration named *MYCONFIG.INI*, use the command `SET CSNTAPI=MYCONFIG`.

You can also use configuration files in other directories, including shared files in remote directories. To use the configuration *CSNTUSER.INI* in a remote directory *P:\MYUSER*, use the command `SET CSNTAPI=P:\MYUSER\CSNTUSER.INI`.

Testing the Configuration

Step 1: Start the Communications Server Node

Start the SNA Node Operations application under the Communications Server program group if it is not already started. Click **Start** on the toolbar to start the local node and specify the configuration file you used earlier in this task.

Step 2: Verify that All Configured Links Are Active

If you configured any connection definitions for Communications Server, such as a host link, ensure that these start successfully by clicking on **Connections** from the toolbar. Any configured links should be shown in the list box. If the links do not become active after a short period of time, ensure that the remote nodes are ready to receive link activation requests from Communications Server.

If your machine is configured as a network node, it may be that a remote node must start a link with Communications Server in order to connect into an APPN network, in which case no link definitions are necessary on Communications Server. Ensure that any remote nodes with links defined to Communications Server have been started.

After Communications Server has been started and all links into the SNA network have been established, you are ready to have your SNA API Client connect with Communications Server.

Step 3: Establish the Connection from the SNA API Client

If you are running an APPC or LUA application locally, a connection to Communications Server will be established when the application is started.

If you have applications on the SNA API Client to which you want to give other SNA nodes access, you need to start the Attach Manager application. Attach Manager automatically starts transaction programs configured to start with Attach Manager.

CPI-C or APPC Configuration

This section describes the steps to configure Communications Server to support CPI-C or APPC applications (for example, APPC3270 or 5250 emulation to an iSeries system). APPC networking is very flexible and versatile and, therefore, it can be complicated. The complexity of this configuration depends on how you will use Communications Server and how much work your application will do for you.

Only the following steps are absolutely required: Step 1 (Node Setup) and Step 2 (Device Configuration). The remaining steps may also be required, depending on your responses to the questions in "Before You Begin."

Read **all** of the documentation that came with your application before proceeding with this configuration. There are a number of questions which you must answer before proceeding, and your application documentation can help you answer them.

Before You Begin

Before you begin this procedure, collect the following information:

- A control point name for this node. The name must be unique within the network.
- The name of the network in which this node will operate.

If you are not using APPN and if you do not know what the value should be, you can make up a fictitious name.

  - **For communications with a zSeries:**
    - A local node ID (XID) corresponding to the host's PU (for switched physical media) *or* a local secondary station address corresponding to the host's PU (for nonswitched physical media)

A local node ID consists of a block ID followed by a PU ID.

    - The LU 6.2 names available on the same host PU and their NAU addresses.
    - Addressing information for contacting the host's communication controller (such as the token-ring, ethernet or TCP/IP address)
- Will this node operate in an APPN network? If so, will it function as an APPN end node or as an APPN network node? This information is used to set the control point type in Step 1.
- Will you run APPC applications (such as 5250 emulators) locally? If so, you do not need to create a Transaction Program definition, and Step 4 is not required.

  However, if you will be configuring applications for other nodes to access, Step 4 is required.
- If you are not connecting Communications Server to an APPN network, and you are establishing a session from the local node to another node in the network (instead of receiving a session request from another node), you need the control point name of the remote node and the LU name on that node to establish your session. Step 5 makes use of this information.
- If you are running an APPC application locally, what does it require in terms of session information? Does it require that you specify a partner LU name and

mode name as parameters (to the application itself, not the Communications Server)? Or does it require that you specify a CPI-C Side Information Profile? If it requires that you specify a CPI-C Side Information Profile, Step 6 is required.

- Will you establish a link from this node to another node, or will another node call into this node? If you need to establish a link with another node, then you must know the addressing information for establishing that link. Use Step 3 to configure that link.
- If you are using AnyNet SNA over TCP/IP:
  - The IP address of your machine
  - The IP address of your gateway or partner
  - The domain name suffix administered by your site administrator.

See the *Network Administration Guide* for more information.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Device configuration
3. Connection configuration
4. Transaction program configuration
5. Partner LU 6.2 configuration
6. CPI-C side information configuration
7. Start the node
8. Verify that the link is active
9. Start the session

3270/LUA applications

This section describes the steps to configure Communications Server for a 3270 session with a host. For APPC 3270 configuration, go to "CPI-C or APPC Configuration" on page 60.

Before proceeding with this configuration, locate **all** the information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin this procedure, collect the following information:

- A local node ID (XID) corresponding to the host's PU (for switched physical media) *or* a local secondary station address corresponding to the host's PU (for nonswitched physical media)

A local node ID consists of a block ID followed by a PU ID.

- At least one local LU address available on the same host PU

- Addressing information for contacting the host's communication controller (such as the token-ring address)
- If you are using AnyNet SNA over TCP/IP:
  - The IP address of your machine
  - The IP address of your gateway or partner
  - The domain name suffix administered by your site administrator.
- If you are using QEL, the user IDs must be configured in the IBMCSAPI group on the Windows system where Communications Server resides.

See the *Network Administration Guide* for more information.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Device configuration
3. Define host connections and LUs
4. Start the node
5. Verify that the link is active
6. Start the application

Focal Point Configuration

Use this section to define remote focal point locations for routing alert information for specific management services applications. The local node will locate remote focal points through an APPN network, so you must configure the local node to operate there; to accomplish this, use "APPN Network Node Configuration" on page 48. You may also use this task to configure the local node as an APPN end node; to accomplish this, select **End node** as the node type and disregard the Partner LU 6.2 Configuration step.

Before proceeding with this configuration, locate **all** required information listed in "Before You Begin." If the following configuration steps do not mention some field names that appear on the configuration panels, you can accept the default values for those fields.

Before You Begin

Before you begin this procedure, collect the following information:

- The control point name of the primary focal point
- The control point name of the backup focal point, if applicable
- The management services category for which you want to forward alert data to the specified focal point
- The applications on the focal points which are used to process the alert data

Before you continue, make sure that the local node is configured to connect into an APPN network as described above. The following steps assume that local node configuration is complete.

Summary of Steps

To begin configuring, follow the steps listed in "Getting Started" on page 44.

This type of configuration requires the following steps:

1. Node setup
2. Device configuration
3. Connection configuration
4. Focal-point configuration
5. Start the node
6. Verify that the link is active
7. Establish a session

Chapter 5. Using Communications Server

Use the SNA Node Operations application of Communications Server to start, stop, and monitor resources in your network.

Starting Communications Server

1. From the Start menu, select **Programs**, **IBM Communications Server**, then **SNA Node Operations**.
2. The Communications Server SNA Node Operations window appears. Either click **Operations** and then click **Start Node**, or click the Start/Stop Node icon.
3. The Open File window appears. Select the appropriate configuration.
4. Click **Open** to start the node.

After the node is started, you can close the Node Operations panel.

Alternatively, you can use the **csstart** command to start Communications Server. See "Command Line Utilities" on page 71 for details.

Stopping Communications Server

To stop all Communications Server connections and any application or emulator sessions that you are supporting through the server:

1. From the Start menu, select **Programs**, click **IBM Communications Server**, and then click **SNA Node Operations**.
2. The Communications Server SNA Node Operations window appears. Click **Operations**, then **Stop Node**, or click the Start/Stop Node icon.
3. Click **OK** at the verification prompt.

Alternatively, you can use the **csstop** command to stop Communications Server. See "Command Line Utilities" on page 71 for details.

Replacing the Default Configuration

You can define more than one Communications Server configuration and then switch between configurations as your needs require.

When you start Communications Server, the default configuration is offered as the first choice for node configuration or node operation unless you specify another configuration. After you complete a new configuration, you can replace the default configuration with your new one.

To replace a default configuration file,

1. Open the configuration file that you want to use as your new default.
2. From the Start Menu, select **Programs**, click **IBM Communications Server**, then click **SNA Node Configuration**.
3. The SNA Node Configuration window appears. Click **File**, then click **Open**.
4. Select the file you want to use as the default.
5. Click **File** and then click **Save**.

© Copyright IBM Corp. 1997, 2007

6. A message will ask you to confirm that you want to use the new file as the default. Click **Yes**.

Use SNA Node Operations to apply your new configuration and ensure that the relevant part of the product runs as you have configured it.

If you have errors in configuration reflected at run time, you can get error messages at this time. Refer to the help for the error message for more information.

SNA Node Operations

SNA Node Operations enables you to display information about Communications Server resources and to manage these resources. The following tasks are supported:

- Starting and stopping resources
- Deleting resources
- Displaying resource information
- Changing session limits
- Adding and modifying select resources

Using the tree view on the SNA Node Operations panel, you can select to manage or display information on the following resources:

- Relational View

  Provides an alternate view of your configuration that shows the hierarchical relationship between sessions, connections, LUs, devices, and data link controls.

- Devices and DLCs

  Data link controls (DLCs)
  Shows how data is formatted for transmission on the physical connection.

  Devices
  Shows the workstation hardware, such as adapters, that is used to transmit and receive data.

- Connections

  Downstream Connections
  Shows connections to downstream nodes

  Host Connections
  Shows connections to the host

  Peer Connections
  Shows link stations to the adjacent nodes.

- Host resources

  DLUR PUs
  Shows dependent logical unit requester (DLUR) physical units (PUs), which are PUs in an APPN end node that own dependent LUs, but request that a dependent LU server provide the SSCP services for those dependent LUs.

  Host Pools
  Displays the host pools that are currently defined.

  Local LU 0 to 3
  Shows local logical units (LUs) type 0, 1, 2, or 3, which are dependent LUs on the workstation that provide services for 3270 terminal and printer emulation applications.

PU2.0/DLUR Gateway
Displays information about SNA downstream workstations and the connections to them. Also shows information about the dependent LU requester services.

- TN3270E server

  TN3270E Ports
  Displays the IP ports used by the TN3270E server and the default pool associated with each port.

  TN3270E Summary
  Displays a summary of settings and client connections for the TN3270E server, including the number of active and inactive connections.

  TN3270E Sessions
  Displays and enables you to stop TN3270E server client connections.

  TN3270E Filter Settings
  Displays information about IP filters that are configured for the TN3270E server.

- TN5250 server

  TN5250 Ports
  Displays the IP ports used by the TN5250 server and the iSeriess associated with each port.

  TN5250 Summary
  Displays a summary of settings and client connections for the TN5250 server, including the number of active and inactive connections.

  TN5250 Sessions
  Displays and enables you to stop TN5250 server client connections.

  TN5250 Filter Settings
  Displays information about IP filters that are configured for the TN5250 server.

- CPI-C and APPC

  Partner LU 6.2
  Shows remote computers that communicate through an APPC session with local LU 6.2s.

  Local LU 6.2
  Shows local logical units (LUs) type 6.2, which are independent or dependent LUs that provide APPC services.

  LU 6.2 Sessions
  Shows sessions that transport data between two partner LU 6.2s. Conversations between transaction programs use LU 6.2 sessions.

  Modes
  Displays names used by the initiator of a session to designate the characteristics desired for the session, such as traffic pacing values, message-length limits, sync point and cryptography options, and the class of service (COS) within the transport network.

  Transaction Programs
  Shows programs that process transactions in an SNA network. There are two kinds of TPs: application transaction programs and service transaction programs.

CPI-C Side Information
Shows a set of parameters that are associated with a specified symbolic destination name.

- APPN options

  Connection Networks
  Shows APPN nodes in a LAN that have direct links with each other without requiring logical link definitions at each node.

  Focal Points
  Shows systems that provide centralized network management services.

  Intermediate Sessions
  Shows APPN network nodes that can pass information, but whose end points are elsewhere.

  Directory Entries
  Shows information on each LU known at the network node, including local LUs, as well as LUs discovered by CP-to-CP sessions to end nodes.

  Topology
  Shows the topology of the SNA network when the network is configured as a network node. Information is provided on all network nodes and links in the network.

  RTP Connections

  Network Node Statistics

- Hot standby

  Critical Servers
  Displays definitions and status information for configured critical servers.

  Backup Servers
  Displays definitions for backup servers.

- AnyNet

  AnyNet IP to LU Mapping
  The IP Address to LU Mapping resource shows attributes used to map an IP address to an LU name.

- Relational View

  Data Link Controls

Using the Menu Bar

From the menu bar, select one of the following:

Operations
Starts or stops an SNA node. To start a node, you must also specify the configuration file you want to use.

Sets or changes session limits (CNOS). Click the appropriate item to manage the session limits on a mode between the local LU and a partner LU.

Applies an additional configuration file to the active node. Select the configuration file you want to use.

Server Connects to a remote Communications Server. TCP/IP or NetBIOS needs to be running and configured so that the client and server can access TCP/IP for using sockets. The TCP/IP hostname, IP address, or machine

name of the server is used to establish the connection with the server. You can connect to several servers at the same time.

You need a user ID and password to connect to the remote server. The user ID must be a member of the IBMCSADMIN user group or have administrator privileges for the operating system to be authorized to administer the server. The user ID you logged on with is used initially to log on to the remote server. If this ID is not authorized, Communications Server prompts you to enter an ID and password.

To connect to a remote Communications Server, do *one* of the following:

- Select **Other** on the Server menu and enter the name of the server to which you want to connect, or
- Select **Other** on the Server menu and click the **Discover** button to discover Communication Servers. The discovered servers will be in a dialog box. You can select one of the servers in the list if you want to connect to it, or
- Choose the multiple node view from the toolbar. All discovered servers are displayed in the left side of the split screen.

To connect to the local Communications Server, choose the **Local** option on the menu.

Launch

Launches other Communications Server programs, such as Log Viewer, Trace Facility, or SNA Node Configuration. For more information on Log Viewer, see "Viewing Communications Server Logs" on page 80. For more information on Trace Facility, see "Obtaining Trace Data on the Server" on page 81. For more information on SNA Node Configuration, see Chapter 4, "Configuring Communications Server for Windows," on page 43.

View Turns off or on the tool bar or status bar of your Communications Server window

Refreshes the display of resources.

Specifies the view displayed: single node, multiple node, or custom group.

Window

Manages the multiple windows within the Node Operations main window.

To open a new window, either choose New Window from this menu or click on the right icon on the tool bar.

Displaying a Resource

To display a resource, select a resource from the hierarchical view on the left side of the split screen. The data is displayed on the right side of the split screen.

Starting a Resource

To start a resource:

1. Select a resource from the hierarchical view on the left side of the split screen.
2. Click the right mouse button on the resource icon on the right side of the split screen to display the action shortcut menu.
3. Click **Start** to start the resource. A started (active) resource can be stopped, but not deleted.

Stopping a Resource

To stop a resource:

1. Select a resource from the hierarchical view on the left side of the split screen.
2. Click the right mouse button on the resource's icon to display the action shortcut menu.
3. Click **Stop** to suspend the resource. If a resource is stopped (inactive), it can be deleted or redefined.

Modifying a Resource

To delete a resource:

1. Select a resource from the hierarchical view on the left side of the split screen.
2. Click the right mouse button on the resource's icon to display the action shortcut menu.
3. Click **Modify** to bring up the configuration window for the resource.

Deleting a Resource

To delete a resource:

1. Select a resource from the hierarchical view on the left side of the split screen.
2. Click the right mouse button on the resource's icon to display the action shortcut menu.
3. Click **Delete** to delete the resource.

A resource to be redefined need not be deleted, but it should be stopped (inactive).

Updating SNA Resources Dynamically

It is not necessary to delete an SNA resource before redefining it. A resource must be inactive, however, when you redefine it. To redefine resources, use the Apply New Configuration selection under the SNA Node Operations **Operations** menu.

If a redefinition is rejected for a reason other than that the resource is active, check the message log for details. Most rejections are due to inconsistencies in the configuration, such as one of the following:

- Attempting to define a link station with the adjacent CP name set to the local CP name.
- Attempting to define two link stations on a non-switched port with the same destination address.

Such rejections occur regardless of whether you are defining a resource for the first time or redefining it.

Keep in mind the following when redefining resources:

- You may not change the ls_role of a port that has link stations defined on it. The link stations must be deleted before the port role can be changed (because a link station may default to the port ls_role).
- You may not change the type of a named DLC. The DLC definition must be deleted before the same name can be reused for a DLC of a different type.

Remote Operations

By default, the administration tools (Node Operations, Node Configuration, Configuration Verification, and Log Viewer) run locally on the Communications

Server machine that is being administered. However, you may use the tools to administer Communications Servers running on remote machines. You can administer remote Communications Servers either from a machine that has Communications Server installed or from a machine that has only the Administration Client installed.

Use the Server menu in Node Operations and Node Configuration to select and connect to a remote Communications Server machine. This remote connection uses the TCP/IP or NetBIOS protocol. All the queries and actions you select are performed against the remote machine.

The remaining administration tools (Configuration Verification and Log Viewer) are not enabled to directly connect to a remote machine. These applications all operate on files, so you need access to the appropriate files on the remote server. You can get access to the configuration files and log files by using the file server capabilities of Windows to share the Communications Server installation directory, then mapping a drive to this shared directory from the administration machine. This enables you to view or update files directly.

Performing Administrative Tasks on the Communications Server

Some administrative tasks for Communications Server can only be completed by a person with a user ID that is contained in the IBMCSADMIN user group or that has administrator privileges for the operating system. Your user ID must be in IBMCSADMIN if you want to:

- Remotely configure and manage the server
- Delete SNA resources through SNA Node Operations
- Stop SNA resources through SNA Node Operations.

During the installation procedure, the IBMCSADMIN group is established and one user ID is placed in this group. Use the Windows User Manager to add additional user IDs to this group as needed. Refer to the Windows documentation for more information on the User Manager.

Command Line Utilities

You can use the following Communications Server commands:

CSDISPLAY resource [object_id] [/D#]

Displays information about Communications Server resources, where:

resource

Is one of the following:

BSV Hot Standby backup servers

CNT Connection networks

CON Connections

CPS CPI-C side information

CRL Critical server connections

CRS Host Standby Critical servers

DCN Downstream connections

DLC Data link controls

DEV Devices

DIR Directory services

DPU DLUR PUs

FPT Focal points

HCN Host connections

INT Intermediate sessions

LU0 Local LU 0 to 3

L62 Local LU 6.2

L6C LU 6.2 conversation

L6S LU 6.2 sessions

MOD Modes

NNS Network node statistics

NOD Nodes

PCN Peer connections

PLU Partner LU 6.2

PDG PU 2.0/DLUR gateways

RTP RTP connections

RTN RTP tuning

TOP Topology

TRP Transaction programs

3FS TN3270E filters

3PT TN3270E ports

3SS TN3270E sessions

3SY TN3270E summary

5FS TN5250 filters

5PT TN5250 ports

5SS TN5250 sessions

5SY TN5250 summary

object_id
Specifies a case-sensitive value (for example, LU001 or LINK001) to search for. Only objects whose first attribute matches this value are displayed.

/D# Specifies the level of detail to display. The # is a number between 1 and 3, where 1 is the lowest level of detail and 3 shows all information. 2 is the default.

CSLIC [number]
Enables you to update the number of concurrent licenses that you have purchased. If `number` is not supplied as a parameter, the following information is displayed:

- Number of licenses purchased
- Number of licenses in use
- Highest number of licenses ever used

CSMODIFY resource object_id action

Enables you to control Communications Server resources, where:

resource

Is one of the following:

CON Connections. You can start, stop, or delete this resource.

CPS CPI-C side information. You can delete this resource.

CRL Hot Standby critical server connections. You can start or stop monitoring this resource.

CRS Hot Standby critical servers. You can start or stop monitoring this resource.

DCN Downstream connections. You can start, stop, or delete this resource.

DLC Data link controls. You can start, stop, or delete this resource.

DEV Devices. You can start, stop, or delete this resource.

DPU DLUR DPUs. You can start or stop this resource.

HCN Host Connections. You can start, stop, or delete this resource.

LU0 Local LU 0 to 3. You can delete this resource unless the LU is in use.

L62 Local LU 6.2. You can delete this resource.

L6S LU 6.2 sessions. You can stop this resource unless a CP-CP session is active.

PCN Peer connections. You can start, stop, or delete this resource.

PDG PU 2.0/DLUR gateway. You can delete this resource, unless a downstream LU is in use.

RTP RTP connections. You can switch the path of this resource.

3SS TN3270E Sessions. You can stop this resource normally.

5SS TN5250 Sessions. You can stop this resource normally.

object_id

Specifies a case-sensitive value (for example, LU001 or LINK001) to search for. Only objects whose first attribute matches this value are displayed.

action One of the following:

SN Normal stop

SI Immediate stop

ST Start

DL Delete

PS Path switch

TH Start Hot Standby monitor

PH Stop Hot Standby monitor

CSNTPD [-s | -q | -d]
Gathers problem determination data. The **-s** flag suppresses registry collection. The **-q** flag sets quiet mode to suppress pop-ups. The **-d** flag collects Dr. Watson User.dmp.

The output files are as follows:

cspddata.exe
Self-extracting file containing the problem determination data

registry.dat
Exported registry data

CSQUERY [-p | -q]
Queries the status of the Communications Server. If the **-p** flag is used, all messages are shown in a message box. Messages are written to the standard output device (usually the screen) by default. The **-q** flag suppresses all output. Along with the running or stopped status, the default and active configuration file names are shown. If the server is running, CSQUERY returns zero, otherwise a non-zero value is returned.

CSSTART [-p | -q] [-a | -m | -d] [cfgfile]
Starts Communications Server with a specified configuration. CSSTART has a user exit capability that enables you to start other processes when the node is started. If there is a batch file named csntstart.bat in the Communications Server directory (usually C:\Program Files\IBM\Communications Server), it is run after the node is started. You can put any command in this batch file, such as a command to start trace utility programs.

If the **-p** flag is used, all messages are shown in a message box. Messages are written to the standard output device (usually the screen) by default. The **-q** flag suppresses all output.

The **-a** flag enables automatic start up after rebooting the machine.

The **-m** flag disables automatic start up.

The **-d** flag makes the configuration file name specified in **cfgfile** the default.

The **cfgfile** parameter is the full path name of the configuration file you want to use to start the product. If no path is specified, the path is assumed to be in the Communications Server installation directory under the private subdirectory (C:\Program Files\IBM\Communications Server\Private, for example). If no file extension is specified, .acg is assumed. If no **cfgfile** parameter is specified, the default configuration file is used. If no default configuration file has been set, an error is reported. If the server is successfully started, CSSTART returns zero; otherwise a non-zero value is returned.

CSSTOP [-p | -q]
Stops Communications Server. If the **-p** flag is used, all messages are shown in a message box. Messages are written to the standard output device (usually the screen) by default. The **-q** flag suppresses all output. If the server is successfully stopped, CSSTOP returns zero; otherwise a non-zero value is returned.

CSTRACE
The CSTRACE command has several parameters that are described in the

following section. Where CSTRACE actions ask you to provide a function ID, component ID, and one or more trace options, refer to Appendix B, "CSTRACE values," on page 87.

CSTRACE APPLY [-f function_ID -c component_ID -o trace_options] [-r] [-t trunc_length] [-m msg_id] [-w] [-n]

Changes trace options immediately, where:

-f function_id
Specifies the function (group) to trace, where **function_id** is an integer. If you specify the **-f** flag, you must also specify the **-c** and **-o** flags.

-c component_ID
Specifies the component to trace, where **component_id** is an integer.

-o trace_options
Specifies the trace options to use, where **trace_options** is a hex value. Where the value has leading zeros, those zeros are optional.

-r Clears the trace buffer.

-t trunc_length
Specifies the maximum trace data length, where **trunc_length** is an integer between 992 and 131072. The default is 16352.

-m msg_id
Stops the trace when the specified message is written into the message log (PCW MSG.MLG).

-w Specifies the size of the trace file.

-n Specifies the number of trace files. Data is written to multiple (**n**) files of size (**w**).

CSTRACE FORMAT[filename]

Converts the trace data to a human-readable file. If you do not specify the file to format, Communications Server formats **nstrc.trc**. If you specify **[filename]**, the file must have the extension **.trc**. The resulting file has the same name as the fn.tr, but has and extension of .tlg.

CSTRACE RESET

Discards the current trace data.

CSTRACE SAVE [-a] [filename]

Saves the current trace data to a file. If you specify the **-a** flag, the data is appended to the file. The default is to overwrite the current trace data. **[filename]** is the name of the file to save.

CSTRACE SHUTDOWN

Shuts down the trace facility.

CSTRACE START [-f function_ID -c component_ID -o trace_options] [-r] [-t trunc_length] [-s storage_number] [-b block_size] [-l file_name] [-m msg_id] [-w size] [-n num_of_files]

Starts the trace facility with the options you specify, where:

-f function_id
Specifies the function (group) to trace, where **function_id** is an integer. If you specify the **-f** flag, you must also specify the **-c** and **-o** flags.

-c component_ID
Specifies the component to trace, where **component_id** is an integer.

-o trace_options
Specifies the trace options to use, where **trace_options** is a hex value. Where the value has leading zeros, those zeros are optional.

-r Clears the trace buffer.

-t trunc_length
Specifies the maximum trace data length, where **trunc_length** is an integer between 992 and 131072. The default is 16352.

-s storage_number
Specifies the number of blocks in the trace buffer.

-b block_size
Specifies the size of a block in the trace buffer.

-l file_name
Specifies the file name, which contains trace options to be enabled.

-m msg_id
Stops a trace when a specific message is issued to the Message Log.

-w size
Writes data to multiple fixed-length files (size in KB). File names starting from nstrc001.trc; cannot be used with **-b** or **-s**.

-n num_of_files
Specifies the total number of trace files. Default value is 999. **-n** is only valid with the **-w** option.

For detailed information on function, component and options, see Appendix B, "CSTRACE values," on page 87.

CSTRACE STATUS
Displays the current active trace.

CSTRACE STOP [-f function_ID -c component_ID -o trace_options]
Suspends one or more active trace options. If you do not specify an option, all active traces are suspended.

-f function_id
Specifies the function (group) to trace, where **function_id** is an integer. If you specify the **-f** flag, you must also specify the **-c** and **-o** flags.

-c component_ID
Specifies the component to trace, where **component_id** is an integer.

-o trace_options
Specifies the trace options to use, where **trace_options** is a hex value. Where the value has leading zeros, those zeros are optional.

SNAFORMAT [flags] [input file]
Formats LLC2 and EEDLC connectivity traces, where flags can be one of the following:

-s Summary output (.sum file)

-d Detailed output (.det file)

-h Skip the HPR flow formatting

The input file must be the formatted file, for example **nstrc.tlg**. The output files are created in the same directory as the input file (fn,sum and fn.sum)

APPNT.bat and APPNF.bat

APPNT.bat is a batch file used to start common traces from the command line. **APPNF.bat** is a batch file used to stop and format traces.

Chapter 6. Problem Determination and Reporting

This chapter describes the information that will help you to solve a problem, to recognize a problem accurately before reporting it to your system administrator, or to report a problem to IBM.

These are the processes that you should follow:

1. Problem determination (PD)
2. Problem-source identification (PSI)
3. Viewing message logs
4. Obtaining trace data
5. Problem reporting (PR)

Problem Determination (PD)

Problem determination (PD) is the first stage in solving your problem.

Symptoms

You recognize there are differences between what you expect to see and what you do see.

Environment

Identify the environment in which the problem occurs:

- Communications Server configuration
  - Communications Server version and CSD level
  - The name of the SNA Node configuration file
- Workstation configuration
  - The machine type and model, the system memory, the video adapter
  - The communication adapter you are using
  - Other adapters (especially communication adapters) installed
  - Other devices installed, such as sound cards, modems, or fax machines
- Software configuration
  - Windows version, level, and fix packs (if any)
  - Communication device-driver version and level
  - Other communication programs (such as Microsoft SNA Server, or Microsoft Data Link Control) that are running and using resources

Problem Type

Categorize your problem into one of the following types:

1. **Installation Problem**. Note the error message and take the appropriate action as described in the help.
2. **Configuration or Setup Problem**. If Communications Server did not connect to the remote node, verify that your hardware, software, and Communications Server configurations are correctly set up for your environment. To help you to isolate what is wrong, look up any error messages in the installed help files. Refer to other sources, such as the *Network Administration Guide*, for detailed information.

© Copyright IBM Corp. 1997, 2007

3. **Operational Problem**. Although you were able to connect to the remote node, you might have difficulty doing some operations or using a certain Communications Server function. Look up any error messages in the online help, and study this manual to see how the operation should be performed.

Problem Area

You also need to identify where the problem is:

- Installation and configuration
- Node Operations
- API
- Other Communications Server functional area

Problem Re-creation

As the final step of the problem determination process, try to re-create the problem. The problem re-creation procedure and the probability of the recurrence of the problem are key factors in isolating it. If the problem is intermittent, any factor that you think is related to it becomes important information.

Problem-Source Identification (PSI)

The second stage is to try to identify the real cause of the problem; this depends largely on the problem area:

1. **Installation and configuration**. Make sure that your workstation hardware and software are configured correctly. You might also need to contact your system administrator to verify that your Communications Server setup values are correct for your system.
2. **Node operations**. Identify the operation you are having a problem with:
   - Starting resources
   - Stopping resources
   - Displaying resources
   - Using the Log Viewer
   - Using the Trace Facility
3. **API**. The cause could be in an application that uses an API, or the problem could be in Communications Server. Identify which of these is causing the problem. The API trace might also help you to identify the source of the problem.
4. **Other Communications Server functional area**. Make sure that you can explain why you think that the source of the problem is in Communications Server.

Viewing Communications Server Logs

The Communications Server Log Viewer utility enables you to view information contained in a Communications Server log. You can use the viewer during problem determination to view the messages in the message log. The default name of the message log output file is PCWMSG; its file extension must be .MLG.

To view messages in a message log:

1. From the SNA Node Operations window, click **Log Viewer**.
2. From the list of logged messages, double-click on a message number to display message help.

Note: The Windows Application Event Log will indicate if there was a problem logged for Communications Server. Start there to determine if the problem lies with Communications Server or another product.

Obtaining Trace Data on the Server

The Communications Server trace facility enables you to log trace information for certain Communications Server functions, which can be useful in identifying the source of problems. After formatting, the trace data is saved to a file. You can view this file with your own editor, then send it to IBM.

To start a trace, follow these steps:

1. From the SNA Node Operations window, click **Trace Facility**. The trace status on the title bar displays the current state:

   Active Trace data is being collected by the trace facility.

   Inactive
   No trace data is being collected. The trace facility is not ready to collect data.
2. From the Trace Facility window that appears, select the type of data you want to trace from the Function Name, Component Name, and Trace Options lists. You can trace multiple functions and components, as well as multiple trace options for the various components.

   Function Name
   A specific set of Communications Server features, such as User Services.

   Component Name
   The name of a specific part of a function, such as Node Initialization (for the User Services function).

   Trace Options
   The options associated with a particular component, such as API trace (for the Node Initialization component).
3. Start tracing data by clicking **Start**, or apply changes to the trace options by clicking **Apply**.
4. Run the operation that you want to trace.
5. Optionally, stop the trace by clicking **Stop**.
6. Save the trace data to your hard disk by clicking **Save**.
7. Click **Format** to specify a formatted trace file name and to format the trace data. Click **OK**.
8. Click **Clear** to clear the trace buffer where you save a trace.

Command line tracing

Alternatively, you can use the capabilities of the following command line batch files:

APPNT
Use **APPNT** to start a trace from the command line.

APPNF
Use **APPNF** to stop or format a trace from the command line.

Obtaining Trace Data on the SNA API Client

The Communications Server trace facility enables you to log trace information for certain Communications Server functions, which can be useful in identifying the source of problems. After formatting, the trace data is saved to a file. You can view this file with your own editor, then send it to IBM.

To start a trace on the Windows SNA API Client, follow these steps:

1. Double-click the trace icon in the client folder.
2. Click **Start** to turn on tracing with the current settings. Tracing begins immediately.
3. Once you have captured the information you want, click **Stop** to end the trace.
4. If you want to format the trace, click **Format**.

Problem Reporting (PR)

If your problem still persists even at this stage, and if you think that the source is in Communications Server, report the problem to IBM.

When you do this, you need to provide the information you obtained through the problem determination (PD) and problem-source identification (PSI) processes:

1. PD information includes:
   - Symptoms
   - Environment
   - Type of problem
   - Problem area
   - Problem re-creatability and re-creation procedure
2. PSI information includes:
   - Error messages (if any)
   - Key factors related to the problem
   - Output from Trace facility
   - Message log
   - Data in the **\PRIVATE** subdirectory, such as .ACG files.

Submitting Problem Determination Information and Receiving Fixes Electronically

Communications Server provides you with utilities for sending problem determination information to a specified destination, as well as receiving fixes electronically from IBM. These utilities include the following:

- Information Bundler
- Information Sender
- Get Fix / Apply Fix
- World Wide Web Access

Information Bundler

This utility gathers system files and specific trace and log files, as well as registry information such as the software installed or running on a machine.

1. To run the utility, select **Start > Programs > IBM Communications Server > Problem Determination > Information Bundler**. A window will open containing information about the utility's progress.
2. The **Information Bundler** creates a file named **cspdata.exe** in the Communications Server installation subdirectory, which by default is C:\Program Files\IBM\Communications Server. This file can be used by the IBM Support Center to aid in problem determination.

See the **CSNTPD** command in "Command Line Utilities" on page 71 for information on gathering problem determination data.

Information Sender

This utility enables you to send a compacted file that contains the system and diagnostic data collected by the **Information Bundler** to an FTP server. To run the utility, select **Start > Programs > IBM Communications Server > Problem Determination Utilities > Information Sender**. This utility will start your default browser and establish a connection to the IBM FTP site. Use your browser to upload the **cspddata.exe** file. If you are using Internet Explorer v5, use drag-and-drop to drag the **cspddata.exe** file from the installation subdirectory, which by default is C:\Program Files\IBM\Communications Server, and drop it into Internet Explorer. If you are using Internet Explorer, select **file > upload** to upload the file.

World Wide Web Access

You can access Communications Server's support pages on the World Wide Web by selecting **Start > Programs > IBM Communications Server > Problem Determination Utilities > Service Information**. The application will attempt to locate a Web browser installed on your system. If one cannot be found, you may specify the location of a specific browser. Click **OK** to launch the browser and access the Communications Server support pages. The support pages will give you access to code fixes, tips, news groups, online documentation, and more.

Contacting IBM

This section lists a number of ways you can reach IBM for various reasons. Depending on the nature of your problem or concern, we will ask you to be prepared to provide us with information to allow us to serve you better.

If you have a technical problem, please take the time to review and carry out the actions suggested here. Use your local support personnel before contacting IBM. Only persons with in-depth knowledge of the problem should contact IBM; therefore, support personnel should act as the interface with IBM.

If you determine that you need to contact IBM, you can do any of the following:

- Consult the **Customer Service and Support Guide**, which is a card contained in the product package.
- Access the Communications Server for Windows Web page at:
  http://www.ibm.com/software/network/commserver/windows
- Access the Communications Server for Windows support page at:
  http://www.ibm.com/software/network/commserver/windows/support

You can manually send files and emails to IBM in the following ways:

- Zip files using the following naming convention:
  `12345.BBB.LLL.CSPDDATA.ZIP`

- FTP to FTP.EMEA.IBM.COM\TOIBM\WINDOWS
- Send email to WEBSPHERE_SUPPORT@MAINZ.IBM.COM

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.

Appendix A. Directory Exploitation

You can configure multiple clients from a central location using the Lightweight Directory Access Protocol (LDAP) to simplify the configuration process. This section provides conceptual information helpful in planning for LDAP.

Client Configuration Concepts

There is an LDAP object type, the server list, to help you configure certain client applications.

The server list object represents a list of servers that can be associated with APPC or 3270 sessions. The name of a server list object can be specified when configuring APPC or 3270 LU data for users, organizational units, or organizations. You can create any number of server list objects.

To minimize duplication of information, most configuration information can be inherited from a higher level. For example, the configuration for a specific user is inherited from the *container*, the hierarchy of the organizational units and organization that comprises a user's distinguished name, in which the user is located. Specific values can be overridden at the user level. Containers inherit from their containers.

Inherited parameters are displayed automatically when the object details are displayed. They can be accepted or overridden at that time.

The SNA API-specific data consists of:

- The APPC configuration attributes for user or container objects, which are used to store client configuration data associated with LU 6.2 applications.
- The 3270/LUA configuration attribute for user or container objects, which is used to store client configuration data associated with 3270/LUA applications.
- The iSeries configuration attribute for user or container objects, which is used to store client configuration data associated with EHNAPPC applications.
- The server list object, which represents a server discovery method to be used by LU 6.2 and 3270/LUA applications. A server list consists of one of the following:
  - A list of explicit servers
  - An implicit list of TCP/IP servers within a scope
  - An implicit list of TCP/IP servers that are not within a scope (they are unscoped)
  - Any server (server name is *)

  The name of a server list object can be specified when configuring LU 6.2 and 3270/LUA data for user, group, or container objects.

Planning for Client Configuration

Before you configure your clients, consider:

- Where to locate each piece of information in the directory tree in order to take advantage of inheritance.

© Copyright IBM Corp. 1997, 2007

- How LUs should be planned in order to maximize reusing the information. If wildcard values and nondedicated LUs are used, less information needs to be configured at the user level. You should weigh this advantage against the needs of the network.

Before configuring client information, gather the following information:

- An explicit list of up to four servers using IP to connect to a Communications Server.
- A scope name. A TCP/IP connection will be established with a Communications Server within the specified scope.
- A single item list with a server name of * (asterisk). A connection will be established with any Communications Server in the network that can satisfy the request.

When you configure a server list, specify the IP host name or Internet address of the Communications Server that you want to connect to.

- The types of applications the SNA API Client will run: APPC and CPI-C, LUA, or EHNAPPC.
- The address of the system where Communications Server resides.
- If you are configuring APPC and CPI-C applications, you might also need the following:
  - If transaction programs on the SNA API Client will need to be run by remote SNA nodes, then you must create a transaction program definition.
  - If you will be running an APPC application program locally, check to see what session information it requires. You may need to specify a partner LU 6.2 name and mode name as parameters, for example. You may also need to configure CPI-C side information.
- If you are configuring the SNA API Client to run LUA applications and require a specific LU address, you need the LU name of an LU configured on Communications Server.
- If you are configuring the SNA API Client to run EHNAPPC applications, you need the list of iSeries hosts that the EHNAPPC applications will access and a list of user IDs and passwords authorized to access these iSeries hosts.

Appendix B. CSTRACE values

Table 8. CSTRACE Values. Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
3	APPN and APPC				
		1	N-BASE		
		2	NOF API		
				1	FLAG IPS trace
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		3	MS API		
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		4	CPI-C API		
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		5	RUI API		
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		6	CSV API		
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		7	APPC API		
				1	FLAG Verb trace only

© Copyright IBM Corp. 1997, 2007

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		11	SLI API		
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
		8	CCE API		
				1	FLAG Verb trace only
				2	FLAG Verb trace (up to 128 bytes of data)
				4	FLAG Verb trace (full data)
4	Connectivity				
		49	AnyNet		
				40	FLAG Internal Calls
				80	FLAG Transport Provider Calls
		33	LAN (LLC2)		
				1	FLAG I-frames only trace
				4	FLAG Truncate to 128 bytes of data
				2	FLAG All frames trace
		9	SDLC		
				10	FLAG DLC Interface
				20	FLAG Outbound Data
				40	FLAG Inbound Data.
				1	FLAG Entry
				8	FLAG Exit
				2	FLAG Control Flow
				80	FLAG Internal Data
		19	LAPB		
				10	FLAG DLC Interface
				20	FLAG Outbound Data

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
				40	FLAG Inbound Data
				1	FLAG Entry
				8	FLAG Exit
				2	FLAG Control Flow
				80	FLAG Internal Data
		20	QLLC		
				10	FLAG DLC Interface
				20	FLAG Outbound Data
				40	FLAG Inbound Data
				1	FLAG Entry
				8	FLAG Exit
				2	FLAG Control Flow
				80	FLAG Internal Data
		22	EEDLC		
				10	FLAG Outbound Data
				20	FLAG Inbound Data
				1	FLAG Entry
				8	FLAG Exit
				2	FLAG Control Flow
				80	FLAG Internal Data
		59	TN3270E Server		
				40	FLAG Internal Calls
				80	FLAG Transport Provider Calls
		60	TN5250 Server		
				40	FLAG Internal Calls
				80	FLAG Transport Provider Calls
		1	Signalling Protocol		
				2	FLAG Control Flow
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Messages Received
				20	FLAG Messages Sent
				40	FLAG Internal Trace
		7	Adapter		
				2	FLAG Control

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Level 1 (least)
				20	FLAG Level 2 (average)
				40	FLAG Level 3 (more)
				80	FLAG Level 4 (most)
		17	MPA Adapter		
				2	FLAG Control
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Level 1 (least)
				20	FLAG Level 2 (average)
				40	FLAG Level 3 (more)
				80	FLAG Level 4 (most)
		4	Buffer Relay		
				2	FLAG Control
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Internal Data 1
				20	FLAG Internal Data 2
		5	Configuration		
				2	FLAG Control
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Instance Data
		6	Connection Manager		
				2	FLAG Control
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Primitives and Messages
				20	FLAG Internal Messages Flows

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
		8	PDLC_DLC		
				1	FLAG Entry
				2	FLAG Control
				8	FLAG Exit
				10	FLAG Internal Trace 1
				20	FLAG Internal Trace 2
				30	FLAG Internal Trace 3
				40	FLAG Internal Trace 4
				80	FLAG Internal Trace 8
		11	V.25bis Signaling		
				2	FLAG Control Flow
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Messages Received
				20	FLAG Messages Sent
				40	FLAG Internal Trace
		12	X.25 Signaling		
				2	FLAG Control Flow
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Messages Received
				20	FLAG Messages Sent
				40	FLAG Internal Trace
				80	FLAG NPDU Trace
		21	OEM Adapter		
				2	FLAG Control Flow
				1	FLAG Entry
				8	FLAG Exit
				10	FLAG Level 1 (least)
				20	FLAG Level 2 (average)
				40	FLAG Level 3 (more)

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
				80	FLAG Level 4(most)
1	User Services				
		1	Node Initialization		
				1	FLAG API trace
				2	FLAG Procedure trace
				4	FLAG DebugStr trace
				8	FLAG Popup trace
				10	FLAG Exhaustive
		2	Configuration API		
				1	FLAG API trace
		3	Message Facility		
				1	FLAG Procedure trace
		4	SNA Node Operations		
				1	FLAG Procedure trace
		5	Automatic Dial Utility		
				1	FLAG Procedure trace
		6	ASCII Configuration		
				1	FLAG Binary conversion trace
5	SNA Client Services				
		1	API Client Services		
				10000000	FLAG Primitive trace (full)
				40000000	FLAG Primitive trace (header only)
				20000000	FLAG Primitive trace (header + 128 bytes)
				0FFFFFFF	FLAG Internal Event trace (ALL)
				0000000F	FLAG Internal Event trace (CM)
				000000F0	FLAG Internal Event trace (NIA)

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
				00000F00	FLAG Internal Event trace (SEC)
				0000F000	FLAG Internal Event trace (SMA)
				00010000	FLAG Internal Event trace (MUX)
				00020000	FLAG Internal Event trace (CSIT)
				00040000	FLAG Internal Event trace (QEL)
				00080000	FLAG Internal Event trace (UTL)
				000C0000	FLAG Internal Event trace (LBA)
8	SLP Services				
		1	SLP Services		
				10000000	FLAG Packet trace
				0FFFFFFF	FLAG Internal Event trace (ALL)
				0000000F	FLAG Internal Event trace (UA)
				000000F0	FLAG Internal Event trace (SA)
				00000F00	FLAG Internal Event trace (NLG)
9	APPN subagent				
		1	IBM APPN subagent		
				10	FLAG VarBind trace data
				10000	FLAG NOF VCB data
				1	FLAG Internal Event trace
				100	FLAG Memory trace
				00FFFFFF	FLAG All data and events
10	RUN COMMAND support				
		1	RUN COMMAND support		
				0x00FFFFFF	FLAG Trace all events and data
				0x00000010	Memory allocations/ deallocations

Table 8. CSTRACE Values (continued). Values for function ID, component ID, and trace options

ID	Function	ID	Component	ID	Trace Option
				0x00000100	FLAG MDS_MU dump
				0x00001000	FLAG VCB dump
11	Hot Standby				
		1	Hot Standby		
				1	FLAG Heartbeat Monitor Trace
				2	FLAG Procedure trace

Appendix C. 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 (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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.

© Copyright IBM Corp. 1997, 2007

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:

IBM Corporation
TL3B/062
3039 Cornwallis Road
RTP, NC 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.

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.

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. 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 IBM's application programming interfaces.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

AIX
AS/400
Advanced Peer-to-Peer Networking

AnyNet
IBMNetView
OS/2
System/370
System/390
S/370
S/390
VTAM

The following product names are registered trademarks or trademarks of Tivoli Systems Inc. or IBM Corporation, in the United States, other countries, or both:

- Tivoli, Manage.
- TME
- Tivoli Ready
- Tivoli Enterprise

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, Windows XP, Windows Server 2003, Windows Vista and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Index

Numerics

A

B

C

D

E

F

G

© Copyright IBM Corp. 1997, 2007

H

I

L

M

N

O

P

R

S

T

U

V

W

Printed in USA

GC31-8424-05