IBM Communications Server for Data Center Deployment
on Linux for System z

Quick Beginnings

Version 7.0

GC31-6769-04

IBM Communications Server for Data Center Deployment
on Linux for System z

Quick Beginnings

Version 7.0

GC31-6769-04

Note:
Before using this information and the product it supports, be sure to read the general information under "Notices," on page 101.

Sixth Edition (December 2012)

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

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

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

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

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

Internet email:
comsvrcf@us.ibm.com

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

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

- Title and order number of this document
- Page number or topic related to your comment

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

Contents

Tables

Figures

About This Book

This book introduces IBM® Communications Server for Data Center Deployment on Linux for System z (CS Linux), program product number 5725-H32, an IBM software product that enables a computer running Linux to exchange information with other nodes in a Systems Network Architecture (SNA) network.

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

CS Linux
CS Linux operates on the following:
- 32–bit Intel workstations running Linux (i686)
- 64–bit AMD64/Intel EM64T workstations running Linux (x86_64)
- IBM pSeries® computers running Linux (ppc64)

CS Linux for System z®
CS Linux for System z operates on System z mainframes running Linux for System z (s390x).

There are two different copies of the *IBM Communications Server for Data Center Deployment on Linux Quick Beginnings* book, one for each of these two installation variants. Please ensure that you have the correct copy of this book for your CS Linux installation. This book applies to CS Linux for System z.

CS Linux provides building blocks for a wide variety of networking needs and solutions. They can be used to exchange information with nodes in SNA networks, or to provide host access for Telnet programs communicating over Transmission Control Protocol/Internet Protocol (TCP/IP).

How to Use This Book

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

Road Map

This book is for management and technical personnel involved in network planning, and for anyone interested in Communications Server for the Linux operating system.

To find the information you need to get started with CS Linux, use Table 1.

Table 1. Getting Started Road Map

If you want to...	Refer to...
Read about CS Linux	Chapter 1, “About IBM Communications Server for Data Center Deployment on Linux,” on page 1
Plan how to use CS Linux in your network	Chapter 2, “Planning for Your Network and CS Linux for System z,” on page 17

Table 1. Getting Started Road Map (continued)

If you want to...	Refer to...
Install CS Linux on Linux servers	Chapter 3, “Installing CS Linux on Linux Servers,” on page 25
Install Remote API Clients on Linux (32–bit Intel, 64–bit Intel/AMD, or pSeries)	Chapter 4, “Installing IBM Remote API Clients on Linux,” on page 37
Install Remote API Clients on Linux for System z	Chapter 5, “Installing IBM Remote API Clients on Linux for System z,” on page 43
Install Remote API Clients on AIX®	Chapter 6, “Installing IBM Remote API Clients on AIX Systems,” on page 49
Install Remote API Clients on Windows	Chapter 7, “Planning for and Installing the Remote API Client on Windows,” on page 53
Configure CS Linux	Chapter 8, “Configuring and Using CS Linux,” on page 65
Find information about the CS Linux documentation and other publications, including online information	Chapter 9, “Information Resources for CS Linux and SNA,” on page 99
Read notices and trademark information	“Notices,” on page 101

Typographic Conventions

The typographic styles used in this document are shown in Table 2.

Table 2. Typographic Conventions

Special Element	Sample of Typography
Emphasized words	**back up files before deleting**
Document title	*IBM Communications Server for Data Center Deployment on Linux Administration Guide*
File or path name	**/usr/spool/uucp/myfile.bkp**
Program or application	**snaadmin**
User input	**0p1**
Computer output	**CLOSE**

Abbreviations Used in This Book

This book uses the following abbreviations:

AIX	Advanced Interactive Executive
API	Application Programming Interface
APPC	advanced program-to-program communication
APPN	Advanced Peer-to-Peer Networking
BrNN	Branch Network Node
COS	Class of Service
CPI-C	Common Programming Interface for Communications
CSV	Common Service Verb
DDDLU	Dynamic Definition of Dependent LUs
DES	data encryption standard
DLC	Data link control
DLUR	Dependent LU Requester

DLUS	Dependent LU Server
FTP	File Transfer Protocol
HPR	High-Performance Routing
IETF	Internet Engineering Task Force
ISO	International Organization for Standards
ISR	Intermediate session routing
LAN	Local area network
LDAP	Lightweight Directory Access Protocol
LEN	Low-entry networking
LLC2	Logical Link Control 2
LU	Logical unit
LUA	Conventional LU Application Programming Interface
MDS-NMVT	Multiple Domain Support—Network Management Vector Transport
MPC	MultiPath Channel
MS	Management Services
NMVT	Network Management Vector Transport
NOF	Node Operator Facility
OSI	Open Systems Interconnection
PU	Physical unit
RFC	Request For Comments
RLE	Run-Length Encoding
SAA	Systems Application Architecture®
SAP	Service access point
SNA	Systems Network Architecture
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol/Internet Protocol
TN	Telnet
TP	Transaction program
VT	Virtual Terminal
WAN	Wide area network

What Is New for This Release

CS Linux Version 7.0 is a follow-on product to CS Linux Version 6.4, which continues to be supported.

New Functions

The following functions have been added to CS Linux in this release:

- Tracking SNA LU resources used by clients on a domain of servers.

  You can track SNA LU resources used by clients that access a pool of LUs. This function dynamically updates the LU records on a server with the system name (DNS information) of the client for each transaction.

  With this capability, clients that are mobile or appear to be mobile from the mainframe host perspective can be accessed from the host. This function also allows queries to display, in real time, who last used a particular LU, and assists in problem determination on connections from workstation to host applications.
- Mapping hardcoded LU aliases on client applications to LU aliases in a domain of servers.

  Some applications are hardcoded to use specific LU aliases. If you are running such an application on two or more Remote API Clients, this function maps these LU aliases to specific LUs on the server (with a different LU for each copy of the application). This allows applications migrating from full-SNA stack implementations to configured for specific LU aliases on the server, and to take advantage of some of the redundant capabilities of the server domain implementation, using all of the resources of the servers in that domain.

- Enhanced support for Enterprise Extender and Connection Networks in IPV6 environments.

  You can use this function to define a Connection Network on IPV6 in which the connections use the IP address to connect, as opposed to using a DNS hostname.
- TN3270 Server and TN Redirector enhancements:
  - Support for 256–bit SSL encryption, providing higher security
  - CS Linux for System z only: Support for the z90Crypto adapter, to improve performance during SSL handshaking

Functions That Have Been Retired

No functions have been retired in this release.

Where to Find More Information

The Bibliography identifies other books in the CS Linux library, as well as books that contain additional information about SNA, and other related products.

Chapter 1. About IBM Communications Server for Data Center Deployment on Linux

This chapter explains how CS Linux is packaged and describes its functions, features, and benefits.

IBM Communications Server for Data Center Deployment on Linux Features and Packaging

CS Linux is communication software that runs on the Linux operating system. It consists of the features described in “IBM Communications Server for Data Center Deployment on Linux” and “Advanced Networking Features” on page 6.

CS Linux on System z is a solution that allows you to consolidate your servers and simplify your networks.

- Servers can be consolidated onto a single hardware platform using the System z technology, which provides benefits such as sharing of processors and memory among virtual servers.
- Dual SNA and IP networks can be simplified to just IP networks, allowing SNA traffic to flow through the IP network to CS Linux for System z, where it can be converted back to SNA and sent across a highly reliable and secure interface to z/OS® CS. An additional benefit of this network simplification is that the need for SNA skills is centralized in one location.

IBM Communications Server for Data Center Deployment on Linux

IBM Communications Server for Data Center Deployment on Linux connects applications across SNA and TCP/IP networks. It converts a System z VM or LPAR running Linux into an SNA node by equipping it with SNA resources and protocols; this enables it to communicate with other computers in an SNA network. It also provides TCP/IP functions to allow IBM Communications Server for Data Center Deployment on Linux to be used within your TCP/IP network or at the boundary between TCP/IP and SNA networks.

Where CS Linux is communicating with an SNA host computer, it can operate in many different ways. Figure 1 on page 2 illustrates two examples of how CS Linux could be deployed:

- In the first example, CS Linux is installed on a separate z800 system to offload the main z/OS system. An Enterprise Extender link using IP, or an LLC2 link, is used to connect the two systems.
- In the second example, CS Linux is installed on one or more VMs or LPARs in the main z/OS system. Although CS Linux and z/OS Communications Server are on the same mainframe, they are two separate SNA nodes, and so an Enterprise Extender link using HyperSockets IP or an LLC2 link is still required between them.

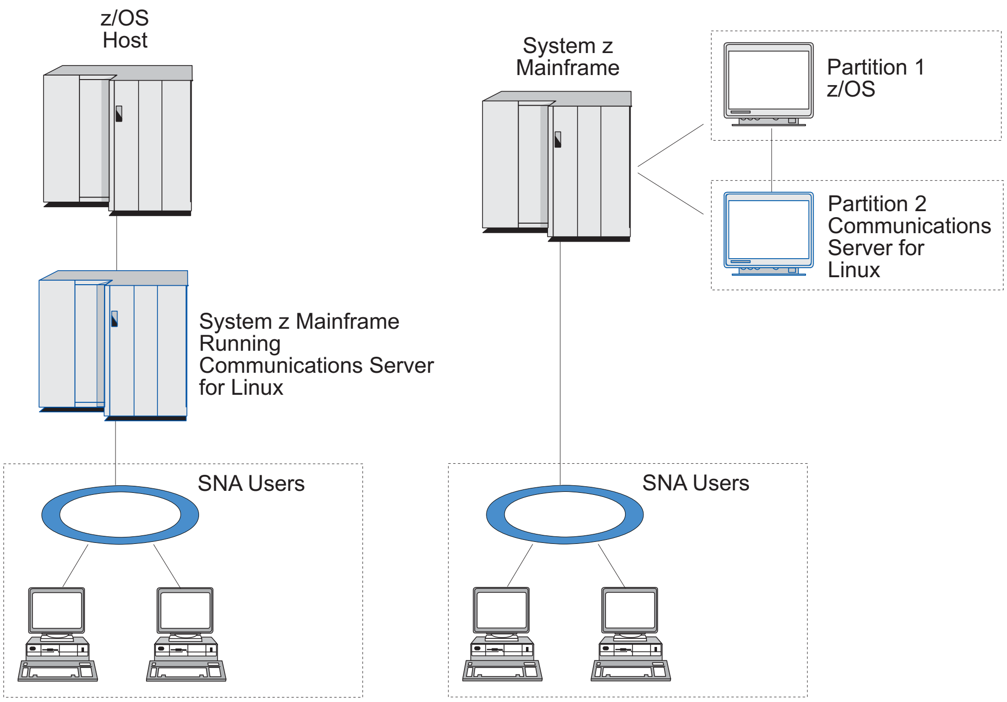


Figure 1. CS Linux for System z Host Communications

The two arrangements illustrated are conceptually the same, and the same CS Linux configuration (including the communications link between CS Linux and the SNA host) is required for both. For clarity, the diagrams in this book show the first arrangement, with CS Linux and the SNA host on separate computers.

CS Linux provides the following services:

Network Support

CS Linux supports subarea and peer-to-peer networks:

SNA Subarea Networks

These networks (also known as host-mediated networks) are hierarchically organized, with one or more host computers controlling communication between computers, managing the network, and providing processing services and high-capacity data storage. All other nodes in the network are dependent on the control of a host.

Linux computers can participate in a subarea network by being configured as host-dependent nodes.

Peer-to-Peer Networks

For distributed processing environments, CS Linux supports APPN networks. In these peer-to-peer networks, Linux computers retain processing functions and communicate directly with each other as peers.

An APPN network consists of peer nodes of the following types:

- APPN network node (which provides traffic control, dynamic route computation and selection services, and network management services)
- APPN end node (which uses APPN network node services to communicate with peer nodes)

- LEN node (which communicates directly with adjacent nodes or nodes configured to appear adjacent)

Note: Host computers can function as peer nodes in an APPN network by using independent LU 6.2 to communicate with Linux computers and other hosts in the network.

Providing Subarea Functions in an APPN Network

The dependent LU requester (DLUR) function enables traffic between hosts and host-dependent nodes to be carried in an APPN network.

Data Link Control Options

At the link level, CS Linux offers different connectivity options to help you meet your network's size, speed, security, and cost considerations. (For a detailed list of the link types supported, see “Installation Requirements” on page 19.) It supports data links for different network types, as follows:

Local Area Networks

For LAN connectivity, you can install the appropriate links to communicate using token ring, standard Ethernet, and 802.3 Ethernet protocols.

Wide Area Networks

CS Linux supports X.25 (QLLC) connectivity. This depends on the OEM adapter support on each platform.

Local Attachment

CS Linux supports Channel-to-Channel Multipath Channel (CTCMPC) connectivity for local attachment (CS Linux for System z only).

IP Integration

If your corporate backbone network is based on IP, you can use the Enterprise Extender (HPR/IP) feature of CS Linux to integrate this with SNA, allowing your SNA applications to communicate over the IP network.

LU Support

Logical units (LUs) are application-specific network resources that reside on each node in an SNA network. Each LU acts as an interface that applications use to access links in order to communicate over the network with partner applications on other nodes.

CS Linux supports different types of LUs for different classes of applications.

- In a subarea network, CS Linux supports dependent LUs, which can be any of the following types:
  - LU 0
  - LU 1
  - LU 2
  - LU 3
  - LU 6.2

  LU 0 supports primitive program-to-program communication, typically used at point-of-sale transactions in retail and banking. LU 2 supports terminal emulation applications that enable the Linux computer to emulate an IBM 3270-family terminal. The other LU types enable applications to participate in distributed processing or to communicate with various printers or interactive display terminals.

CS Linux supports host systems that use dynamic definition of dependent LUs (DDDLU), a host feature that enables dependent LUs on the SNA system to be added to the host configuration when the communication link from the SNA system to the host is established. With DDDLU, LUs do not have to be configured statically at the host. (You must still define dependent LUs on the CS Linux node.) This reduces the initial configuration required at the host, and makes later expansion easier.

CS Linux can communicate with both DDDLU-capable and non-DDDLU-capable hosts, with no difference in the configuration required. When the communications link from the CS Linux node to the host is established, a DDDLU-capable host informs the node that it supports DDDLU; the node then sends the required information to define the dependent LUs that use the link. If the host is not DDDLU-capable, CS Linux does not send this information; it assumes that the LUs have already been defined statically at the host.

- Independent LU 6.2 supports independent traffic in APPN networks. Independent LU 6.2 supports autonomous communication and network management, as well as distributed processing.

  In addition, the DLUR function of CS Linux enables traffic from dependent LUs to travel over an APPN network.
- Primary RUI support provides the ability for a CS Linux application to manage downstream LAN/WAN attached dependent LU devices as though it were a mainframe. This function has some restrictions for connectivity, but allows applications to pass data between dependent LU devices without the need for a full mainframe application.

Session Support

A session is a temporary logical channel between partner LUs. Ordinarily, partner applications associated with each LU communicate over the session. CS Linux can support thousands of sessions. CS Linux can also support U-shaped sessions (also known as "local/remote transparency"), in which both primary and secondary LUs reside in the same Linux computer. This enables you to develop and test a pair of source and target transaction programs in one computer without requiring a link connection.

The data flowing on a session between two partner LUs may be compressed, to reduce the bandwidth required.

- For LU type 6.2, CS Linux allows you to specify the use of compression in the configuration of the mode that the session uses. You can specify different compression algorithms to be used, each of which provides a different level of compression (RLE, LZ9, or LZ10). You can also specify different compression levels for data flowing in different directions on the session, or specify compression in one direction but not the other.
- For LU types 0–3, CS Linux allows you to specify the use of compression in the configuration of the link station or PU that the session uses. RLE compression is used for the inbound direction, and LZ9 for the outbound direction.

API Support

CS Linux includes application programming interfaces (APIs) for developing applications for certain types of LUs, for distributed processing, for network management, and for administration of CS Linux itself. CS

Linux provides a range of APIs that are compatible with the APIs provided by members of the Communications Server family running on other operating systems.

An API is an interface that enables a transaction program (TP) to communicate with its supporting LU. It consists of a library of verbs (also called functions, calls, and subroutines) from which the TP selects those it needs to pass to its LU to request an action, such as SEND_DATA. The LU, in turn, processes the verbs and builds a data stream according to the appropriate protocol, appends a header indicating the destination address, and sends the data over the link to partner LUs.

Common Programming Interface for Communications (CPI-C) is one of the most powerful of the APIs because of its portability. Introduced to support dependent and independent LU 6.2, CPI-C complies with Systems Application Architecture (SAA) mandates to unify different platforms and operating systems. CPI-C uses a set of syntax rules that is common to all systems. It has thus become a standard.

As well as the standard C-language CPI-C API, CS Linux also includes a CPI-C API for use by Java applications. For more information, refer to *IBM Communications Server for Data Center Deployment on AIX or Linux CPI-C Programmer's Guide*. In the CS Linux books, all references to CPI-C include Java CPI-C unless stated otherwise.

Other CS Linux APIs include:

- APPC API for peer-to-peer communications between application programs using LU 6.2. The API has the option of being nonblocking. When a TP uses nonblocking verbs, the API can return control to the TP before the requested action has completed. Later, the TP is informed when the action has completed.
- LUA API for communications with host applications.
- CSV (Common Service Verb) API for utility functions such as character translation and application trace control.

In addition, CS Linux includes the following proprietary programming interfaces:

- MS (Management Services) API for network messaging functions.
- NOF (Node Operator Facility) API for applications that configure and manage CS Linux resources.

For more detailed information about an API, refer to the programming guide for the API (see the Bibliography).

Client/Server Support

Computers running CS Linux can be configured to communicate using client/server protocols. When client/server protocols are used in a network, all the computers using client/server protocols to communicate in that network are referred to as a “domain.”

The computers running CS Linux in a client/server configuration can take the following roles:

- A server contains an SNA node and its associated connectivity components. The server provides SNA connectivity to applications on the local system or on other computers in the CS Linux domain. Servers must be Linux systems.
- A Remote API client does not contain SNA node components, but accesses them through a server. A client can access one or more servers

at the same time, and can run concurrent applications as needed. Clients can be running AIX, Linux, or Windows. (A Linux computer can be either a server or a client, but not both; you cannot install both the server and the client on the same computer.)

Servers and clients communicate across the CS Linux domain using TCP/IP. Alternatively, they can communicate using HTTPS via a WebSphere® server, which uses security certificates to authenticate the client connections. You will normally want to use HTTPS if the clients connect across a public network.

In a domain with multiple CS Linux servers, one server holds the master copy of the CS Linux domain configuration file. This server is known as the master server. You can define other servers in the domain to be backup servers, or leave them as peer servers. The domain configuration file is copied to backup servers—either when they are started, or when the master copy is changed—so that all backup servers hold a copy of the latest information. A peer server obtains domain configuration information from the master server as required, but cannot act as a backup server.

If the master server fails, the first backup server on the list of servers defined for the domain takes over as the master. The domain configuration file on this server is used as the master copy, and is copied to other servers as necessary. When the master server is restarted, it receives a copy of the domain configuration from the backup server currently acting as master, and then takes over as the master.

Support for Distributed Applications

In a Client/Server CS Linux system, applications running on Remote API Clients cooperate with connectivity resources on servers to execute a single task. Applications running on other (non-CS Linux) computers can also cooperate with applications on CS Linux computers to perform distributed processing.

CS Linux supports distributed applications using APPC (also known as LU 6.2).

Advanced Networking Features

Included in the CS Linux base product is a set of features that add advanced networking capabilities to it. These features include the following:

- SNA gateway connects LANs to subarea SNA networks.
- Primary LU support provides support for controlling downstream dependent LU devices in the same way as a host mainframe application.
- Branch Extender simplifies large APPN networks by separating out resources in different locations (for example in separate branches of a large organization). This reduces the amount of topology information that must be stored, while still allowing efficient resource location.
- APPC Application Suite provides selected applications for use in APPN networks.
- Enterprise Extender (EE, also known as HPR/IP) allows SNA traffic to be transported natively over IP networks.
- TN Server provides host access over SNA to TN3270 and TN3270E clients, referred to collectively as TN3270 clients.
- TN Redirector provides passthrough TCP/IP host access to TN3270, TN3270E, TN5250 and VT clients, referred to collectively as Telnet clients.

- Remote API Client server provides redundant and high availability access via cloud-like support for remotely distributed Windows, AIX and Linux clients

Note: Advancements in highly available network adapters and operating system platforms concentrate on IP networking features. CS Linux provides SNA over IP connectivity with Enterprise Extender, TN3270 Server, TN redirector and Remote API Client server. To take full advantage of the these advancements, consider using these features when modernizing SNA networking. Legacy network connectivity using SNA over LAN or WAN connections can only take advantage of some of the new advancements in highly available networking devices.

SNA Gateway

A gateway is a user-transparent device that connects dissimilar networks or computer systems, supporting both of the environments that it connects. End users perceive each other as residing in the same network.

SNA gateway enables a CS Linux computer to act as a gateway that links multiple downstream computers in an SNA network to one or more host physical units (PUs), as illustrated in Figure 2. To simplify host connectivity, and to eliminate excess links, SNA gateway acts as a PU concentrator—it treats the multiple computers as a single PU (that appears to reside on the SNA gateway node) and communicates with the host over a single physical connection.

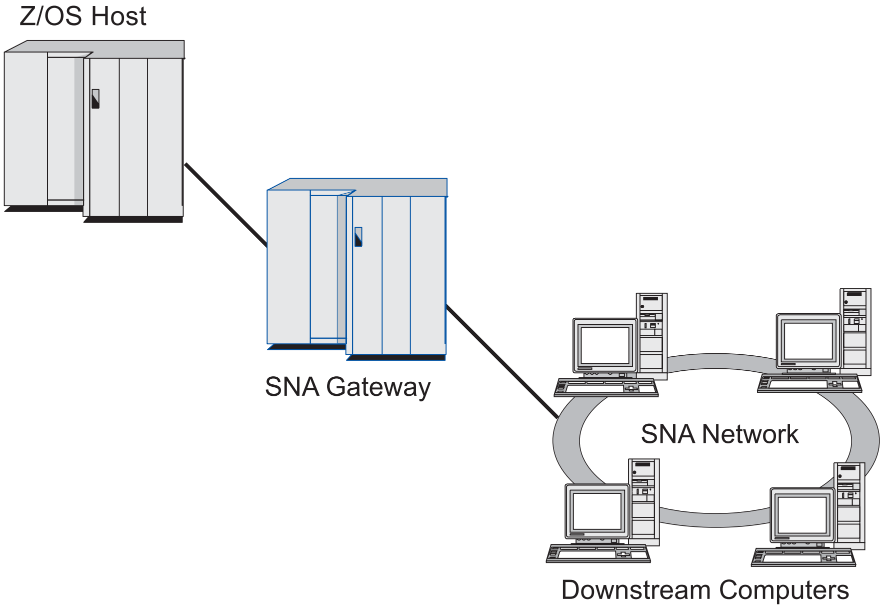


Figure 2. SNA Gateway Linking Multiple Downstream Linux Computers to a Host Computer

Primary LU Support

Primary LU support allows a Linux application to control downstream dependent LU devices as though it were a host mainframe application.

LUA applications usually connect to host mainframes as secondary LUs, so that the host application controls the definition for the sessions and is responsible for sending the BIND to start a session. CS Linux also includes the ability to act as a primary LU to downstream dependent SNA devices over a LAN, using the Primary RUI interface. Using this interface, an application can connect downstream dependent LU sessions without the need for a host mainframe.

To use Primary LU applications, the node must be configured with downstream LUs (or a Downstream PU template) using a host LU name of #PRIRUI#. This indicates to the server that the applications using Primary RUI will control these PUs and the LU resources assigned to them. The PUs can be used on both LAN and WAN ports. Refer to *IBM Communications Server for Data Center Deployment on AIX or Linux LUA Programmer's Guide* for information about programming applications to use Primary RUI.

Branch Extender

Network nodes in an APPN network need to maintain topology information (about the location of other nodes in the network and the communications links between them), and to forward this information around the network when the topology changes. As the network grows in size, the amount of stored information and topology-related network traffic can become large and difficult to manage.

It is possible to avoid these problems by separating the network into subnetworks, so that each node only needs to maintain topology information about the nodes in its own subnetwork. However, this results in increased network traffic when trying to locate resources in other subnetworks.

The Branch Extender feature of APPN, illustrated in Figure 3, provides a solution to these problems.

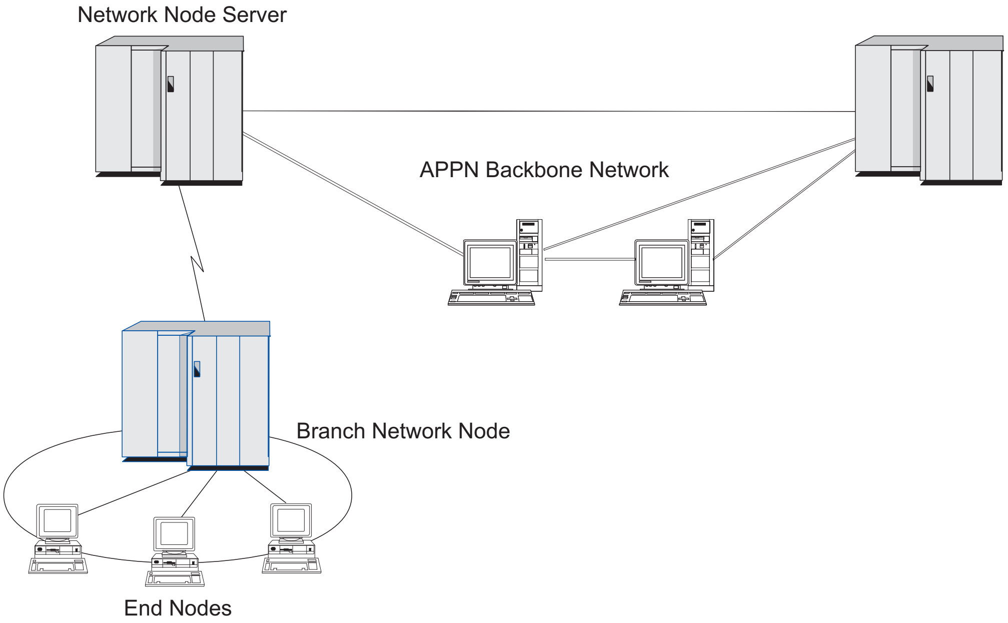


Figure 3. Branch Extender

As the name implies, Branch Extender is designed for networks that can be divided into distinct areas such as separate branches of a large organization. It works by separating out branches from the main backbone APPN network (for example, the network in the organization's headquarters).

Each branch contains a node of a new type called Branch Network Node (BrNN), which is connected to a Network Node in the main APPN backbone network. The BrNN combines the functions of an APPN network node and an APPN end node.

- To the backbone network, the BrNN appears as an End Node, connected to its Network Node Server (NNS) in the backbone network:
  - The nodes in the backbone network are not aware of the nodes within the branch, reducing the amount of topology information that must be stored.
  - Because the BrNN appears as an End Node, it does not receive topology information from the backbone network (topology information is transmitted only between Network Nodes).
  - The BrNN registers all resources in the branch with its NNS as though they were located on the BrNN itself. This means that the nodes in the backbone network can locate resources in the branch without having to be aware of the separate nodes in the branch.
- To the branch network, the BrNN appears as a Network Node, acting as the NNS for End Nodes in the branch. Each node in the branch sees the rest of the network as being connected through its NNS, in the same way as for a standard NNS.

APPC Application Suite

APPC Application Suite is a set of applications that demonstrates the distributed processing capabilities of APPN networks, and can be helpful in configuration verification and problem determination. APPC Application Suite can be used to provide support for operations such as file transfers, which are frequently performed across a network.

APPC Application Suite contains the following applications:

- **ACOPY** (APPC COPY)
- **AFTP** (APPC File Transfer Protocol)
- **ANAME** (APPC Name Server)
- **APING** (APPC Ping)
- **AREXEC** (APPC Remote EXECution)
- **ATELL** (APPC TELL)

These applications can be accessed from a server or from a Linux or Windows client.

Enterprise Extender

Enterprise Extender (also known as HPR/IP) provides a mechanism for integrating SNA applications with an IP network.

SNA applications are designed to use SNA protocols to communicate over SNA networks with other SNA applications. When installed in a TCP/IP network using Enterprise Extender, SNA applications can still communicate; the Enterprise Extender function provides a mechanism for transporting SNA protocols over the IP network. In particular, it provides APPN High-Performance Routing (HPR) functionality, giving the applications the benefits of both APPN and IP connectivity.

Enterprise Extender in CS Linux is implemented simply as a communications link. To connect two SNA applications over IP, you define an Enterprise Extender link, in the same way as for any other link type such as SDLC or Ethernet.

TN Server

3270 emulation programs that communicate over TCP/IP (rather than over an SNA network) are referred to as "TN3270 programs" (Telnet 3270 emulation programs).

TN3270 programs can also include support for TN3270E (Telnet 3270 standard extensions). TN3270E supports 3270 device emulation (including both terminals and printers) using Telnet. It enables a Telnet client to select a particular device (by specifying the LU name or the name of an LU pool), and provides enhanced support for various functions, including the ATTN and SYSREQ keys and SNA response handling.

Note: This guide uses the term TN3270 for information that applies equally to the TN3270, TN3287, and TN3270E protocols.

CS Linux TN server provides access to 3270 host computers for TN3270 users on other computers. TN server enables TN3270 users to share a host connection with CS Linux or with other TN3270 users, instead of requiring a direct link. TN server also enables TN3270 users to access hosts that are not running TCP/IP.

The CS Linux TN server function is illustrated in Figure 4.

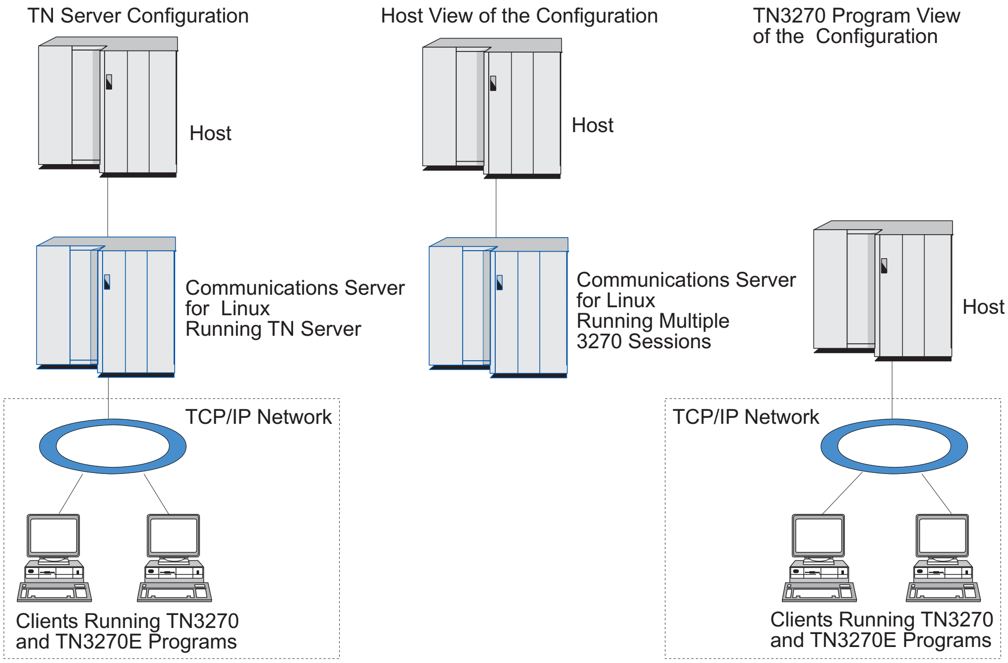

TN Server Configuration
Host
Communications Server for Linux Running TN Server
TCP/IP Network
Clients Running TN3270 and TN3270E Programs
Host View of the Configuration
Host
Communications Server for Linux Running Multiple 3270 Sessions
TN3270 Program View of the Configuration
Host
TCP/IP Network
Clients Running TN3270 and TN3270E Programs

Figure 4. TN Server

The CS Linux TN server feature provides an association between a TN3270 user and CS Linux 3270 LU. All data from the TN3270 user is routed to the LU. This means that the configuration for both the host and the TN3270 user is as though they were connected directly; neither needs to be aware that data is being routed through TN server.

CS Linux TN server supports all TN3270 client emulation programs that correctly implement the protocols defined in IETF RFCs 1123, 1576, 1646, 1647, and 2355.

Security Features: CS Linux TN Server supports data encryption, server authentication, client authentication, and Express® Logon, using Secure Sockets Layer (SSL) software:

- Data encryption means that the data flowing between the TN Server and the TN3270 emulator is in encrypted form.
- Server authentication allows a TN3270 client to verify that the TN Server it is connected to is the one it expects.
- Client authentication allows a TN Server to verify that the TN3270 client connecting to it is the one it expects. The TN Server can also check a revocation list on an external directory server to ensure that the client's authorization has not been revoked.
- Express Logon operates in conjunction with client authentication to remove the requirement for TN3270 clients to provide a user ID and password when connecting to the host. Instead, the client's security certificate is used to retrieve the necessary user ID and password information.

TN Redirector

The CS Linux TN Redirector feature provides passthrough services for 3270, 5250 or VT sessions over TCP/IP. The Telnet user communicates with CS Linux over a TCP/IP connection; CS Linux then communicates with the host over another TCP/IP connection.

CS Linux TN Redirector supports data encryption, server authentication, and client authentication, using Secure Sockets Layer (SSL) software, in the same way as for TN Server for 3270. This allows you to use Secure Sockets Layer (SSL) security checking where necessary, and not on the complete user-to-host connection. For example:

- If clients are connecting to CS Linux over a TCP/IP LAN where checking is not required, but are connecting to a remote host that requires SSL, you can use SSL over the TCP/IP connection between CS Linux and the host. This means that security is checked once for all clients, and individual clients do not have to provide security information.
- If CS Linux is installed on the same site as the host, but clients are connecting in from external sites, you can use SSL over the client connections to CS Linux without having to install SSL software on the host.

Features and Benefits

CS Linux has features and benefits that range from simplifying configuration to improving problem diagnosis to enhancing network performance.

Versatile Building Blocks

CS Linux supports most environments and node functions. In any type of network, subarea or APPN, it enables the Linux computer to function as any one of, or a combination of, the following:

- Host-dependent node
- Peer node (for a description of APPN peer nodes, see the discussion of peer-to-peer networks in “IBM Communications Server for Data Center Deployment on Linux” on page 1)
- Partner (either source or destination) in distributed applications
- Gateway node that interconnects SNA networks

Through network management APIs, the Linux computer can also be configured to function as a Management Services (MS) entry point to provide support for distributed network management. At the link level, the Linux computer can be connected to various LANs and WANs by using any of the supported link types (described in “IBM Communications Server for Data Center Deployment on Linux” on page 1 and “Installation Requirements” on page 19).

Client/Server Operation

Client/server configuration provides the following benefits:

- Concentrating SNA resources on servers reduces the load on clients, improving client performance and minimizing the storage needed to provide SNA services to clients.
- A single data link can be shared by multiple users on different machines, eliminating the need for each machine to have a physical SNA network connection.
- Multiple servers can provide redundant connectivity (for example, by having multiple servers providing access to the same host). Having multiple paths to an SNA resource enables load balancing across the different servers and provides immediate backup in the event that a particular server or link fails.
- By using LU pools across multiple servers, the administrator can easily configure and add servers and users.
- Having fewer links and PUs for host connectivity reduces the size of the host VTAM® definition.
- Administration utilities can be used to configure and manage both node resources (for any computer in the domain) and shared resources. The client/server support provided by CS Linux administration tools enables transparent administration of all domain resources from any computer in the domain.
- SNA applications can be connected over Internet Protocols using TCP/IP and HTTPS for traversing firewalls and for authentication and security.
- Client applications can be mobile, and the Communications Server can track the user so the host can contact the application whereever the application moves.
- Remote users that run application on a “farm” of Windows Terminal Servers (WTS), Citrix servers or UNIX X-servers can connect from any server in the farm. The host application will be able to connect back to the remote user, even as the user switches between using different servers.

Easy Configuration

CS Linux is designed with configuration options and capabilities that reduce configuration time and network complexity. For example:

Motif Administration Program

The easiest way to define and modify the CS Linux configuration is to use the Motif administration program (**xsnaadmin**). This program provides a graphical user interface from which you can view and manage CS Linux resources. This program also simplifies configuration by exposing only the fields whose values typically vary from one installation to another, using default values for other fields.

The Motif administration program includes help screens that provide overview information for SNA and CS Linux, reference information for CS Linux dialogs, and guidance for performing specific tasks.

Dynamic Configuration in APPN Networks

Configuring a node or network is also made easier through the APPN network's dynamic configuration. For example, APPN end nodes and applications dynamically register configuration data to support LU 6.2 sessions, thus making session configuration optional. Further, by having the node control point act as the default local LU, you can avoid LU 6.2 configuration altogether.

APPN also supports dynamic link station configuration in the absence of configured link stations.

Additional User Interface Choices for Administration

The Motif administration program is the recommended interface for configuring and managing CS Linux. However, you have a choice of interfaces for CS Linux, which enables you to work with the one that suits your equipment, needs, and preferences.

Command-Line Administration Program

The command-line administration program (**snaadmin**) can be used to issue commands to manage individual CS Linux resources. You can use **snaadmin** either directly from the Linux command prompt or from within a shell script.

NOF API

The CS Linux NOF API provides the same management functions as the command-line administration program, providing an interface suitable for use within a program (rather than a command script). You can use the NOF API to write your own application programs to administer CS Linux.

Better Performance

CS Linux enhances the inherently high performance of SNA networks and uses class of service operation. CS Linux also optimizes network speed through SNA data compression for LU 0–3 session data, and through different traffic-management methods that balance traffic flow according to network size:

- In APPN networks, CS Linux supports both High-Performance Routing (HPR) and intermediate session routing (ISR), and provides connection network options. While ISR works efficiently for small networks, it degrades the performance of larger ones.
- For larger networks using LAN connectivity options (such as Token Ring or ethernet) or using Enterprise Extender, you can also use the connection network option to improve communication efficiency. The connection network option creates a communications path directly between nodes. This enables traffic to bypass intermediate network nodes.
- Another traffic control mechanism, adaptive session-level pacing, automatically adjusts congestion by regulating the rate at which LUs send message units to partner LUs.

Security Options

With networks becoming more complex and moving to an open architecture, security emerges as a major issue. In SNA networks running CS Linux, you can protect your assets by defining various levels of security through configuration and by implementing certain types of links. For example:

- In a client/server system, you can set up a WebSphere server to provide HTTPS access from Remote API Clients to the servers. This means that the client connections are authenticated using security certificates. (This feature requires some additional software in addition to the standard CS Linux product. See “Installation Requirements” on page 19 for more information.)

- LU 6.2 users can define up to three levels of security—session, resource, and conversation. Respectively, one ensures that the proper LUs are engaged in a session, one restricts access to all applications associated with a particular LU, and one restricts access to a particular application. Additional security is possible through data encryption routines.
- CS Linux TN Server and TN Redirector can provide data encryption, server authentication, and client authentication between the CS Linux server and TN3270 or Telnet clients, using Secure Sockets Layer (SSL) software. (This feature requires some additional software in addition to the standard CS Linux product. See “Installation Requirements” on page 19 for more information.)

Network Management Flexibility

CS Linux supports the Multiple Domain Support-Network Management Vector Transport (MDS-NMVT)network management scheme, which can work as a centralized, distributed, or hierarchical management scheme. It is based on a focal point/entry point architecture that gives you a high degree of flexibility.

Focal points are control nodes that manage the network according to the data they collect from entry points (management applications that reside on all other nodes in the network).

- In centralized management, a single focal point acts as a control point for the entire network.
- In distributed management, several focal points share in managing the network.
- In hierarchical management, focal points are nested according to function.

MDS-NMVT can thus be adapted to manage subarea, standard APPN, and very large APPN networks.

Reliability, Availability, and Serviceability

To help you maintain reliable system operation, CS Linux provides a range of display functions and problem-diagnosis tools.

- The Motif administration program provides enhanced configuration and management tools, including the following:
  - Immediate updates of configuration information
  - Status information for links, sessions, and node resources
- Query and status commands provide you with information about:
  - LU-LU sessions
  - APPN intermediate sessions
  - Active links
  - APPN topology databases, which store link information
- Problem-diagnosis tools are available to help you during the different stages of configuration and operation. They include the following:
  - Diagnostics information collection tool (**snagetpd**) to enable you to collect service information easily
  - Status and error messages to help resolve problems with configuration routines and system operation
  - Logs for collecting network error, failure, and audit information
  - Trace facilities for gathering and formatting detailed problem information

Other utilities help you test for link connectivity and communication between applications.

CS Linux also includes the Management Services API, which is used to develop tools for sending and receiving network alerts and problem data.

All of these management and problem-diagnosis tools are fully integrated into the CS Linux Client/Server model, so you can manage the entire CS Linux domain or collect diagnostics information from a single point in the network.

Network Integration, Growth, and Change

To support network integration, growth, and change, the CS Linux APIs can be used to develop applications for a particular LU, platform, or operating system as your business needs dictate. CPI-C is an especially important API because it is consistent across different platforms and operating systems. It is used to develop applications that can run on any system.

Enterprise Extender also provides a mechanism for integrating SNA and TCP/IP networks.

Chapter 2. Planning for Your Network and CS Linux for System z

This chapter provides an overview of the stages of planning a network that runs CS Linux. It also summarizes the functions that you can configure for the Linux computer and presents guidelines for estimating the resources required to support the functions.

Stages of Network Planning

This section presents some general guidelines for planning, configuring, and managing networks through the use of various CS Linux and Linux utilities.

Planning a network involves balancing function, performance, resources, and cost. Although there is no single best plan for a network, some general guidelines and techniques can help ensure that your plan meets your needs. To plan a network, perform the following tasks:

- Determine the functions the network should provide (such as file transfer or 3270 emulation) and your performance requirements.
- Determine how CS Linux can be configured to provide the functions you need.
- Estimate the resources needed to install CS Linux, to support your performance and capacity requirements, and to support CS Linux functions, and identify the associated costs.

Identifying Functional Requirements for the Network

To determine the functions your network should provide, you need to consider the following questions:

- Is the network to be APPN?
- Will CS Linux run as a client/server system? If so, will all computers operate in a single CS Linux domain, or do I need to define two or more separate domains?
- Do I need more than one server in the CS Linux domain to provide load balancing for connectivity resources? If so, which server will be the master configuration server? Do I need to provide one or more backup configuration servers?
- Do I need to support Remote API Clients connecting to CS Linux servers using HTTPS?
- Will user applications run on the server or on Linux client computers?
- Will the server provide connectivity resources for Windows applications (such as API transaction programs) running on Windows clients?
- Is each server to be an end-point for sessions or is it to be one of the following types of gateways?
  - APPN
  - LU 0
  - LU 2
  - TN Server or TN Redirector
- What types of physical links will the network use?
- Will CS Linux need to support IPv4, IPv6 or both types of connections?

The answers to these kinds of questions help you determine which CS Linux functions the network requires.

Determining How to Configure CS Linux

To determine how CS Linux is to function, you should first decide how work is to flow through the network. The following questions should be considered:

- What resources (such as applications) should be available through the network?
- How many users need access to remote resources?
- How frequently is each resource accessed?
- How can users get access to the network?
- How will user requests be routed through the network?

You can configure CS Linux to support many functions, including, for example, the following:

- APPN network node for intermediate session routing (ISR)
- APPN end node (which communicates autonomously with adjacent nodes but uses APPN network node services to communicate with nonadjacent peer nodes)
- Low-entry network (LEN) node (which communicates directly with adjacent nodes or nodes configured to appear adjacent)
- Use of LU 0, LU 1, LU 2, LU 3, and LU 6.2 (dependent and independent)
- SNA gateway connection to a host

One or more functions can be configured for a given node, depending on your needs. For example, you can configure CS Linux as an APPN network node to provide routing services and ISR, and use the same node for SNA gateway to route dependent LU sessions, such as LU 0 and LU 2. Similarly, you can configure CS Linux to run TN Server and support a shared database, as well as an independent LU 6.2 connection to MQSeries® on the host.

Identifying Resource Requirements for Installation and Operation

To estimate support for the functions of CS Linux, the following questions must be answered:

- What personnel skills do I need?
- What transport media do I anticipate using?
- What are the installation requirements for the configuration I select?
- How much memory and paging space do I need for operation?

Answering these questions helps you identify the types of resources that CS Linux uses when it is configured to support any one or more of the many functions described in "Identifying Functional Requirements for the Network" on page 17. Answering the questions also helps you to understand the relationship between CS Linux functions, Linux resources, and network resources.

How you allocate resources to nodes determines how the network will perform.

Personnel Requirements

Installing, operating, and tuning CS Linux requires the following personnel:

- Network administrators, who plan the network, add new devices, and maintain or enhance overall network performance

- System administrators, who install and maintain CS Linux and the hardware on which it operates, and who configure systems for network connection
- Programmers, who develop customized applications such as transaction programs or network management routines

Network and system administrators should be thoroughly familiar with the hardware on which CS Linux operates and with the Linux operating system. They must know the networks to which various systems are connected and understand SNA concepts in general. They should also be familiar with the following:

- The Motif interface
- **rpm**, the installation tool for Linux
- TCP/IP, if they plan to use Client/Server functions, TN server or Enterprise Extender
- The Windows XP, Windows 2003 Server, Windows Vista, Windows Server 2008, Windows 7, or Windows 8 operating system, if the CS Linux system includes Remote API Clients on Windows
- WebSphere Application Server, if the CS Linux system includes Remote API Clients that connect to servers using HTTPS

Programmers who develop customized applications for SNA should be experienced with the C language (or Java if they are using Java CPI-C), and should be familiar with the APIs that are available in CS Linux.

Transport Media

CS Linux might need to share the underlying transport medium (such as SDLC, token ring) with other communication protocols. Therefore, physical layer bandwidth requirements must accommodate all the protocols and applications sharing the transport medium.

Note: CS Linux can share the token ring and Ethernet adapters with other protocols such as TCP/IP. You might need to specify unique service access point (SAP) addresses for each protocol to use.

Installation Requirements

The functions that you assign to CS Linux (from “Identifying Functional Requirements for the Network” on page 17) also determine the installation requirements. This section provides an overview of the computer resources required for installing CS Linux. For more information, see the documentation supplied with each product (see the Bibliography for a list of the books).

Hardware

CS Linux requires a 64–bit zSeries®, System z9®, or later computer.

Use the command **uname -m** to verify your target computer. The following table shows the appropriate hardware and the response from the **uname -m** for this hardware.

Server Type	Hardware	uname response
System z	64–bit zSeries or System z9	`s390x`

Linux operating system

CS Linux supports the following Linux variants. For up-to-date information about specific version numbers and kernel builds that are supported for each variant, and any additional requirements for specific versions, please refer to the **README** file on the CS Linux install image.

- RedHat Enterprise Linux 5 for System z (RHEL5-s390x)
- RedHat Enterprise Linux 6 for System z (RHEL6-s390x)
- SUSE Linux Enterprise Server 10 for IBM Mainframe (SLES10-s390x)
- SUSE Linux Enterprise Server 11 for IBM Mainframe (SLES11-s390x)

The Security Enhanced Linux (SE Linux) feature on Linux distributions allows Linux to monitor and control access to certain code libraries and functions. Depending on the level of Linux distribution that is installed, there will be some configuration edits needed to enable CS Linux and Linux Streams. To see what these steps are, refer to the CS Linux support web page at http://www-947.ibm.com/support/entry/portal/Overview/Software/Other_Software/Communications_Server_for_Linux for information about how to configure your system for SE Linux.

Link hardware

Link hardware is required only on a server, and not on a client.

CS Linux can be used over TCP/IP interfaces using Enterprise Extender, over a virtual MultiPath Channel (MPC) interface using the Linux for System z MultiPath Channel device driver (CS Linux for System z only), over 802.2 Ethernet or Token Ring OSA connections using the Linux **lcs** device driver (which requires an OSA2 in OSE **chpid** mode), or over 802.2 Ethernet OSA connections using the Linux **qeth** device driver with layer2/VSwitch support.

Additional Software: Linux

CS Linux requires the following additional software. See the **README** file on the CS Linux install image for more details of specific version requirements (depending on your Linux variant), and for instructions on installing these software packages.

- Linux Streams (LiS).
- OpenMotif (required only on a server, and not on a client). This is required in order to use the Motif administration program, which is the recommended method of configuring and managing CS Linux.
- Java (required if you want to use Java CPI-C). You will need the Java Runtime Environment (JRE). If you need to compile new Java classes for use with a Java CPI-C application, you will also need the Java SDK.
- GSKIT. This is required on a server if you intend to use TN Server or TN Redirector with the SSL feature, and on a client if it will connect to CS Linux servers using HTTPS.
  - For a server, the GSKIT software is included on the installation media and is installed automatically when you install CS Linux. For CS Linux for System z, you can optionally use the z90Crypto adapter to provide SSL security: refer to the file **CSLinux_Crypto_settings.pdf** on the installation media for more information about how to configure CS Linux to use this adapter.
  - For a client, see the **README** file on the installation media for more information about obtaining and installing the GSKIT software.
- The Linux for System z MultiPath Channel device driver (CS Linux for System z only; required only on a server,and not on a client. This component is required if you will be using MPC connections to connect to VM/VTAM systems).

WebSphere Application Server (for HTTPS access)

If you will be running a client/server system in which Remote API Clients connect to CS Linux servers using HTTPS, you will need to run WebSphere Application Server to provide HTTPS access from these clients to the servers.

CS Linux operates with WebSphere Application Server Version 5, which can be installed on a computer running any operating system supported by WebSphere. (If necessary it can be installed on the same Linux computer as a CS Linux server.) Refer to the WebSphere Application Server documentation for more information about installing it. You will also need to install an additional CS Linux plug-in on this computer to use WebSphere with CS Linux, as described in “Configuring WebSphere Application Server” on page 31.

Memory and Storage

To support a full range of configurations and services, CS Linux needs the minimum memory required by the Linux distribution plus 64MB, and 200 MB of disk space. In addition, 250 MB of temporary storage is required during installation.

If you decide to install the documentation for CS Linux in softcopy (PDF) form, you need additional fixed-disk storage. To install all softcopy books, you need 80 MB of disk space.

Note: Memory and fixed-disk requirements for other licensed programs, user applications, and data are not included in these requirements; carefully review all system, memory, and fixed-disk requirements with your IBM representative or authorized industry remarketer.

IPv4 and IPv6 Addressing

Computers running CS Linux Version 7.0 can use either IPv4 or IPv6 addresses, with the following constraints.

- All servers in a Client/Server domain must use the same addressing format (IPv4 or IPv6).
  - If the servers use IPv4, clients must also use IPv4.
  - If the servers use IPv6, clients can use either IPv6 or IPv4.
- For TN Server, if CS Linux uses IPv4, TN clients connecting to the TN Server must also use IPv4. If CS Linux uses IPv6, TN Clients can use either IPv6 or IPv4. By default the TN Server accepts connections from both types of clients, but you can configure it to listen on a particular IP address (using the *listen_local_address* parameter in the command-line administration program or a NOF application) in order to restrict it to one type of client connection.
- For TN Redirector, if CS Linux uses IPv4, both TCP/IP connections (from the client to CS Linux and from CS Linux to the host) must also use IPv4.

  If CS Linux uses IPv6, the TCP/IP connection from the client to CS Linux follows the same rules as for TN Server. The connection from CS Linux to the host can use either IPv6 or IPv4. There is no requirement for the two connections to use the same addressing format.
- For Enterprise Extender (HPR/IP), the ports at both ends of a link must use the same addressing format (IPv4 or IPv6).
  - If CS Linux uses IPv4, it can connect only to remote systems that are configured to support IPv4.
  - If CS Linux uses IPv6, you can configure it to use either IPv4 or IPv6 on an Enterprise Extender link. The option you choose must match the configuration at the remote system.

In addition, all links on the same Enterprise Extender port must use the same addressing format (IPv4 or IPv6). If you need to support links with different addressing formats, you must use separate ports. Similarly, all Enterprise Extender ports on the same connection network must use the same addressing format.

To check whether a CS Linux server is running IPv4 or IPv6, use the command **ifconfig -a** and look at the IP address or addresses in the output. These will be IPv4 dotted-decimal addresses, IPv6 hexadecimal addresses, or both. For a Remote API Client on Windows, the equivalent command is **ipconfig** (with no command-line options).If you need to change the computer's IP addressing format, refer to your operating system documentation.

If you are upgrading an existing CS Linux system to Version 7.0 as described in “Migrating from previous levels of CS Linux” on page 26, and you also want to change to IPv6 addressing, you can do the two processes in either order. However, you cannot use the new IPv6 addressing capabilities in Version 7.0 until both processes are complete.

- For a Client/Server system, you must change all the servers in the domain from IPv4 to IPv6 at the same time; do not attempt to run a mixed domain of IPv4 and IPv6 servers.
- Because the upgrade to CS Linux Version 7.0 also requires you to upgrade all the servers at the same time, you may choose to make the change to IPv6 addressing at the same time as upgrading each server. Alternatively, you can change all the servers to IPv6 either before or after the upgrade to Version 7.0, whichever is more convenient.
- After all the servers have been changed to use IPv6 addressing, you can change Remote API Clients to use IPv6 addressing as required. IPv4 clients can continue to operate with IPv6 servers, so there is no need to change all the clients at the same time.

If you are installing a new CS Linux system, you can install it with only IPv6 addressing on all servers and clients if appropriate, or you can use IPv4 addressing initially and then move to IPv6 later(subject to the restrictions above for Client/Server domains).

Configuring your firewall to allow Enterprise Extender, client/server and TN3270 Server traffic

The default filters for the Linux server's firewall settings, in **/etc/sysconfig/iptables**, will prohibit Enterprise Extender, client/server, TN3270 Server and TN Redirector ports from receiving inbound traffic. To enable connectivity over these ports while the firewall is enabled, you can update the **/etc/sysconfig/iptables** file as follows.

Find the section in the file where INPUT statements are listed. All of the statements below should be added to this section.

To enable receiving Enterprise Extender packets, add the following statements identifying the port numbers that Enterprise Extender uses:

```
-A INPUT -p udp --dport 12000 -j ACCEPT
-A INPUT -p udp --dport 12001 -j ACCEPT
-A INPUT -p udp --dport 12002 -j ACCEPT
-A INPUT -p udp --dport 12003 -j ACCEPT
-A INPUT -p udp --dport 12004 -j ACCEPT
```

To allow Remote API Clients to connect to the server, add the following statement identifying the client/server port number:

```
-A INPUT -p tcp --dport 1553 -j ACCEPT
```

To enable receiving TN3270 Server and TN Redirector traffic, add a statement similar to the following for each port number used by a TN Server or TN Redirector; these port numbers are specified using **define_tn3270_access** or **define_tn_redirect**. The example is for the port number 6023.

```
-A INPUT -p tcp --dport 6023 -j ACCEPT
```

After updating the file, you need to restart the iptables service so that the firewall filter is active with the new settings. Use the following command:

```
service iptables restart
```

The firewall will now allow CS Linux to receive Enterprise Extender, client/server and TN3270 Server packets to establish connectivity with partner servers and clients.

Naming Conventions

You can use network IDs to logically segment your physical network. Also, if you plan to connect to other networks, it is highly recommended that you register your network IDs to avoid network name conflicts.

You can define network and LU names as follows:

Network names

You can define different network names (network IDs) to provide segmentation of APPN networks. Segmentation limits the size of network topology databases and the frequency of broadcast LOCATE requests through each network.

To ensure the uniqueness of a network ID, a network administrator can register the network's ID with the IBM worldwide registry. The IBM registry ensures that each network ID is unique among those registered with it. Registry standards are consistent with Open Systems Interconnection (OSI) standards, including OSI country codes, as established by the International Organization for Standards (ISO). For more information about registration, see *User's Guide for SNA Network Registry*.

LU names

You can use wildcards for LU names to minimize system definition and network searches.

Chapter 3. Installing CS Linux on Linux Servers

This chapter provides general information on the steps you will need to take in order to install CS Linux on a Linux server. For detailed step-by-step instructions on the installation process, see the **README** file on the CS Linux install image, which provides up-to-date, detailed information that is specific to your Linux variant. The **README** file includes:

- Exact package names and/or version numbers for the additional software packages that you need
- Details of the installation and setup commands.

How the CS Linux Licensed Program Is Packaged

The CS Linux Licensed Program is delivered as four CD images containing the following.

CD #1: Quick Start
This CD contains a full set of PDF documentation for CS Linux (in the directory **/DOCS**).

CD #2: Communications Server for Linux platforms
This CD contains the files required to install a server on Linux or Linux on System z:

- **README** files containing information about any product changes following the publication of the CS Linux documentation
- Installation scripts
- Install images for the server and for the PDF manuals

CD #3: Communications Server for AIX platforms
This CD contains the files required to install a server on AIX.

CD #4: Remote API CLients
This CD contains the README files, installation scripts and install images required to install each type of client:

- Linux client for 32–bit Intel (i686)
- Linux client for 64–bit AMD64/Intel EM64T (x86_64)
- Linux client for pSeries (ppc64)
- Linux client for System z (s390x)
- AIX client
- 32–bit Windows client
- x64 Windows client

See the **README** files for full details of the files included in the CD images.

Note: For storage requirements, see “Installation Requirements” on page 19.

Preparing for CS Linux Installation

Installing Pre-Requisite Software

Before installing CS Linux, you need to install the pre-requisite software listed in “Installation Requirements” on page 19:

- LiS Streams
- OpenMotif
- Java
- The Linux for System z MultiPath Channel device driver (CS Linux for System z only; required only if you will be using MPC connections to connect to VM/VTAM systems).

For full details of the software packages required and step-by-step instructions for installing them, see the **README** file on the CS Linux install image.

Displaying Product Installation Details

You can display information about CS Linux and related software packages that are already installed. To list all the installed packages, use the following command:

rpm -q -a

To view more details of a specific package, use the following command:

rpm -q -i *packagename*

packagename is the base name of the installed package, for example **ibm-commserver**.

Changing the Language Environment Variable

When you use CS Linux, make sure that the LANG variable is set correctly to indicate the language you want to use.

Use the following command to change the LANG variable:

export LANG=*language*

Replace *language* with the identifier for the language you want to use, which can be one of the following:

Identifier	Language
en_US	English (United States)
ja_JP	Japanese (PC)
de_DE	German
es_ES	Spanish
fr_FR	French
ko_KR	Korean
pt_BR	Portuguese
zh_CN	Chinese (Simplified EUC)
zh_TW	Chinese (Traditional)

Migrating from previous levels of CS Linux

Considerations

If you are upgrading to CS Linux Version 7.0 from an earlier version of CS Linux, you need to consider the following.

1. If you are running CS Linux in a Client/Server configuration with two or more servers, you are advised to upgrade all the servers to Version 7.0 at the same time, before upgrading the Remote API Clients.
   - While you are in the process of migrating the servers, you will not be able to use the Motif administration program or command-line administration program on a back-level server to view and manage resources on a server running Version 7.0.
   - Version 6.3.1.0 and later versions of the Remote API Client will work with an existing server, but only if the operating system on the client is not configured to use IPv6.
   - Earlier versions of the Remote API Client will work with CS Linux Version 7.0.
2. Several data structures in the NOF API were modified in Version 6.2.3.0 to accept the longer address formats required for IPv6 addresses. This means that, if you use any of the following verbs and/or indications in an existing NOF application that was created before Version 6.2.3.0 (even if you are not using the new IPv6 addressing capabilities), you will need to recompile the application to use it with CS Linux Version 7.0.
   - DEFINE_LS, DEFINE_PORT, QUERY_LS, QUERY_PORT when used with an Enterprise Extender (HPR/IP) LS or port
   - DEFINE_TN3270_ACCESS, DELETE_TN3270_ACCESS, QUERY_TN3270_ACCESS
   - DEFINE_TN3270_EXPRESS_LOGON, QUERY_TN3270_EXPRESS_LOGON
   - DEFINE_TN3270_SSL_LDAP, QUERY_TN3270_SSL_LDAP
   - DEFINE_TN_REDIRECT, QUERY_TN_REDIRECT_DEF
   - QUERY_LU_0_TO_3 (for any LU type)
   - TN_REDIRECTION_INDICATION
3. If you want to use the new IPv6 addressing capabilities of Version 7.0, you need to ensure that the CS Linux servers are configured to use IPv6 addressing. See “IPv4 and IPv6 Addressing” on page 21 for more details.

Migration process

If you already have an earlier version of CS Linux installed and are now migrating to Version 7.0, you need to take the following steps:

Save any customized configuration files

If any of the files listed below exist, save them to a temporary directory. Not all of these files will exist in all installations.

/etc/opt/ibm/sna/sna_node.cfg
/etc/opt/ibm/sna/sna_domn.cfg
/etc/opt/sna/sna.net
/etc/opt/ibm/sna/sna_tps
/etc/opt/ibm/sna/ibmcs.kdb
/etc/opt/ibm/sna/ibmcs.sth
/etc/opt/ibm/sna/ibmcs.rdb
/etc/opt/ibm/sna/ibmcs.crl

In addition, if you customized the startup file **/etc/rc.d/init.d/snastart** as described in “Enabling CS Linux” on page 68 to remove the **sna start** command, so that CS Linux is not started automatically at system startup, make a note of the changes you made to this file.

Uninstall the old release

Use the following commands to stop CS Linux and uninstall it. Depending on which previous version is installed and how you installed it, not all of the RPM packages listed may exist on your system.

```
sna stop
rpm -e CS-LINUX-ptf
rpm -e CS-LINUX-Docs
rpm -e CS-LINUX-ecl
rpm -e CS-LINUX
rpm -e ibm-commserver-ptf
rpm -e ibm-commserver-docs
rpm -e ibm-commserver-ecl
rpm -e ibm-commserver
rpm -e gsk5bas
rpm -e gsk6bas
rpm -e gskssl32 gskcrypt32
```

Uninstall LiS

Use the following commands to uninstall the current level of the LiS open source package.

```
PATH=$PATH:/sbin
unset LD_PRELOAD
rmmod streams
cd /usr/src/LiS
make uninstall
make very-clean
cd /usr/src
rm -rf LiS*
```

Change PATH and other environment variables

If you modified any of the following environment variables for the earlier CS Linux version, you should remove the changes you made, because the paths may be different for the CS Linux Version 7.0 Program Product.

```
PATH
LD_LIBRARY_PATH
LD_RUN_PATH
LD_PRELOAD
CLASSPATH
```

You may want to use the **env** command to check all environment variables for references to **sna**:

```
env | grep sna
```

Other packages

You are recommended to check your Java installation and update it to the latest level if necessary.

Final clean-up

The following commands will remove any remaining items from the old level of the product.

```
rm -rf /etc/opt/ibm/sna /var/opt/ibm/sna /opt/ibm/sna
```

Install the new level of the CS Linux Version 7.0 Program Product

Follow the instructions in this manual and in the **README** file to install the product.

Restore the saved configuration

If you saved any configuration files in the first step of this process, now is the time to restore them. First stop the CS Linux software with the following command:

```
/opt/ibm/sna/bin/sna stop
```

If you saved any **ibmcs.*** files in the first step of this process, remove **all ibmcs.*** files from the **/etc/opt/ibm/sna** directory now. For example, if you saved **ibmcs.kdb** and **ibmcs.sth**, you need to remove **ibmcs.crl** and **ibmcs.rdb** even though you do not have saved files to replace them. It is important that you do not run with a mix of saved and new files.

Restore the files you saved to the **/etc/opt/ibm/sna** directory.

In addition, if you have saved changes to the startup file **/etc/rc.d/init.d/snastart**, make the same changes to the new copy of the file as described in "Enabling CS Linux" on page 68 to ensure that CS Linux is not started automatically at system startup.

Now start the CS Linux software again with the following command:

```
/opt/ibm/sna/bin/sna start
```

Installing the CS Linux Licensed Program

Installing CS Linux

If you have a previous level of CS Linux already installed, follow the steps in "Migrating from previous levels of CS Linux" on page 26 to remove it before installing this new level.

To install CS Linux, take the following steps.

1. Copy or FTP the **ibm-commserver-7.0.0.0–s390x.tgz** file from the CD-ROM to the Linux System z system. Ensure that you use binary mode to copy or FTP the file.
2. Log into the Linux System z system as root.
3. Uncompress and unpack the tar file into an empty temporary directory:

   ```
   mkdir /tmp/ibmcs
   cd /tmp/ibmcs
   zcat ibm-commserver-7.0.0.0–s390x.tgz | tar —xf —
   ```
4. Run the **installibmcs** shell script:

   ```
   ./installibmcs
   ```

   This shell script tests for certain pre-requisites and issues warning messages if they are not met. It also prompts you for the following. When you have responded to the prompts, the shell script installs the **rpm** packages.
   - Confirmation that you have read and accepted the CS Linux license terms.
   - The name of the server that will be the master server in the CS Linux domain. If you want to run CS Linux as a standalone node, do not specify this parameter; in this case, the node will not support client/server functions.

   You can override these prompts by specifying additional parameters on the **installibmcs** command as described below.

If the shell script encounters an error that prevents successful installation of CS Linux, it writes an error message to standard output (typically to the screen). For help with resolving any such errors, refer to the **README** file on the CS Linux install image.

For systems with limited memory, you may need to reboot after installing CS Linux before the SNA node can be started. For larger systems this may not be required. If the CS Linux node fails to start, check the **/var/log/messages** file for an entry such as:

kernel: SNA Trace Driver can only get *X* **blocks of memory — please reboot**

If these messages persist even after rebooting, you need more memory.

5. Add the CS Linux binary directories to your PATH. You may want to change your profile to do this automatically:

   export PATH="$PATH:/opt/ibm/sna/bin:/opt/ibm/sna/bin/X11"

   export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib

   export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib

   For Java CPI-C applications you should also set the following environment variable:

   export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar

   For some applications you may also need to set the `LD_PRELOAD` environment variable, but you should not make this a global change in your profile:

   export LD_PRELOAD=/usr/lib/libpLiS.so

6. Start CS Linux. Note that after installation this will happen automatically when the machine is rebooted.

   cd /

   sna start

7. Run the CS Linux Motif administration program. You are strongly recommended to use this program until you are familiar with CS Linux operation.

   You will need to use a remote XWindows server, because the Linux System z system includes only XWindows client capability. On the XWindows server, use the following command:

   xhost +*XXXX*

   XXXX is the TCP/IP name or address of the Linux System z system.

   Now tell the xsnaadmin client where the server is and start it:

   export DISPLAY=*YYYY:Z*

   xsnaadmin &

   YYYY is the TCP/IP name or address of the XWindows server, and *Z* is the virtual display number (typically `0`).

If you need to do an unattended installation, you can provide additional parameters on the **installibmcs** command to confirm acceptance of the CS Linux license terms and to specify the name of the master server. In this case, the shell script will run without prompting for any additional information. Use the following command:

./installibmcs license_accepted [*master_name*]

master_name is the name of the master server. If you want to run CS Linux as a standalone node, do not specify this parameter; in this case, the node will not support client/server functions.

CS Linux online documentation

Follow the steps in “Installing CS Linux” on page 29 to unpack the **tgz** file, and then run the **installibmcsdocs** shell script:

```
./installibmcsdocs
```

Configuring WebSphere Application Server

If you will be running a client/server system in which Remote API Clients connect to CS Linux servers using HTTPS, you will need a computer running WebSphere Application Server to provide HTTPS access from these clients to the servers, as described in “Installation Requirements” on page 19.

This section describes how to set up WebSphere for use with CS Linux:

- Setting up a secure certificate on the WebSphere server that will be presented to clients
- Configuring WebSphere Application Server to work with CS Linux
- Installing the server configuration file on the WebSphere server

You will also need to set up the client security certificate and the client network data file on each Remote API Client to access the WebSphere Application Server. For more information, see the chapter on installing the appropriate client type.

Setting up the WebSphere Application Server's secure certificate

Refer to the WebSphere Application Server documentation for instructions on setting up a secure certificate on the server. This is the server's certificate that will be presented to a Remote API Client during the authentication process when it tries to connect using HTTPS.

You are recommended to configure WebSphere so that it enforces client authentication; see the WebSphere Application Server documentation for more information. This means that WebSphere will request security certificates from Remote API Clients during the authentication process, and will accept an incoming connection from a Remote API Client only if it can verify the authenticity of the client's certificate.

Configuring WebSphere Application Server

To configure WebSphere Application Server to operate with CS Linux, take the following steps. Refer to the WebSphere Application Server documentation for more information.

1. Copy or FTP the two files **snahttpsrv.ear** and **snahttpsrv.cfg** from the **ibm-commserver-https** directory on the Remote API Client installation CD to a directory on the computer where the WebSphere administration console runs, or to a network directory that can be accessed from this computer.

   If the administration console is running on Windows, copying the files is not necessary because you can access the files direct from the CD. You just need to insert the Remote API Client installation CD into the Windows computer's CD drive.
2. Start the WebSphere administration console.

3. Follow the WebSphere documentation to create a virtual host that is accessible only via an SSL secured connection. This virtual host will be used for the Java plug-in that manages SNA HTTPS connections.
4. From the menu bar, choose Applications, Install New Application.
5. Specify the location of the **snahttpsrv.ear** file. Choose the Next button.
6. When prompted to specify a virtual host name in the first two screens, enter the name of the virtual host that you have set up for HTTPS. For all other parameters you can accept the default options unless you need to use any specific WebSphere configuration; choose the Next button on the following dialogs until it is replaced by a Finish button, and then choose the Finish button. The screen should then show **Application installed successfully**.
7. Click on Save to Master Configuration, and then click on the Save button.
8. From the menu bar, choose Applications, Enterprise Applications.
9. Find **SnaHttpTransport** in the list of applications, click on the checkbox next to it, and click on the Start button to start the application. (After this, the application will be started automatically when WebSphere Application Server is started.)
10. From the menu bar, choose Environment, Update Web Server Plugin, and click the OK button. This updates the WebSphere configuration.

Installing the server configuration file

To operate with CS Linux, the WebSphere Application Server requires a list of the CS Linux servers that will be accessed using HTTPS. Create and install this list using the following steps.

1. In the WebSphere administration console menu bar, choose Environment, Manage WebSphere Variables.
2. Look for the **USER_INSTALL_ROOT** variable in this list, and note its value (which is the path of a directory on the WebSphere server). The list of environment variables may span two or more pages, so you may need to use the Next button to scroll through the list.
3. Copy the **snahttpsrv.cfg** file from the location where you saved it in “Configuring WebSphere Application Server” on page 31 (or from the installation CD) into the directory specified by the **USER_INSTALL_ROOT** variable, and then edit this file using a text editor to include a list of CS Linux servers that can be accessed by Remote API Clients using HTTPS. Each server must be specified on a separate line of the file, in the following format:

   server=*servername.domainname.com*

Post-Installation Procedures

This section explains how to perform maintenance tasks that may be required after installing CS Linux.

Client/Server Operation

After installation, CS Linux initially operates as a standalone server (with all components on a single Linux system). If you want to run it as a server in a client/server domain, refer to the chapter on Managing CS Linux Client/Server Systems in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for instructions.

Post-Install Cleanup

When you are done with the installation, you can erase the **tgz** file and temporary directory that were created during the installation process.

Viewing PDF Books

The manuals included on the installation media for this product are in Portable Document Format (PDF). Softcopy format enables you to search, browse, or print the information easily, using hypertext links for related information. It also makes it easier to share the library across your site, because PDF viewers are available for many different platforms.

If you choose to install the PDF manuals when installing the product, they are installed in the directory **/opt/ibm/sna/docs**. The manuals are also included in the directory **/DOCS** on the CS Linux installation media and in the **tgz** file.

You can read the PDF manuals using any PDF viewer, such as Adobe Acrobat on Windows or **xpdf** on Intel Linux.

Reviewing Current Release Information

The latest update of the **README** file for the product, contained in the **/opt/ibm/sna** directory, contains information about any product changes following the publication of the CS Linux library. This file is also included in the root directory of the CS Linux installation media and in the **tgz** file. Review the **README** file whenever you receive product updates.

Configuring SSL for use with TN Server or TN Redirector

If you intend to use TN Server or TN Redirector with the SSL feature, you need to configure the SSL software after you have installed CS Linux.

The SSL software requires two components:

- A key pair is required to allow data encryption and decryption to be carried out.
- A certificate is required to allow server authentication.

The certificate and key pair make up a single record in a keyring database, which is stored on the CS Linux server running TN Server or TN Redirector. CS Linux uses the database to implement SSL.

In order to manage the keyring database, OpenMotif must be installed. See the **README** file on the CS Linux install image for instructions on installing OpenMotif.

To manage the keyring database, type the following command at the Linux command prompt:

```
snakeyman
```

The **snakeyman** command launches a Java program. See the help provided with this program for further instructions.

Each record in the database is identified by a unique name known as a label. If you have two or more records to use on different TN Server or TN Redirector sessions, you need to make a note of the labels you assign when setting up the database; these labels are used to identify which record is to be used on each session. You can also identify one of the records as the default, so that sessions will use this record unless you explicitly specify the label of a different record.

After using **snakeyman** to update the server certificates, you need to exit the **snakeyman** program and then stop and restart the CS Linux node in order to use the updated certificates. Use the following commands to stop and restart the node:

```
snaadmin term_node
snaadmin init_node
```

Backing Up CS Linux Configuration Files

CS Linux automatically backs up the node, domain, TN3270 device data (**tn3270dev.dat**), and TP configuration files whenever you make changes that affect those files (using any of the CS Linux administration tools). For example, when you make a change that affects the node configuration file (**sna_node.cfg**), CS Linux creates a backup file named **sna_node.bk***n*, where *n* is either 1 or 2:

- The first time you change the file, the existing configuration is saved to **sna_node.bk1**.
- The second time you change the file, the existing configuration is saved to **sna_node.bk2**, leaving **sna_node.bk1** unchanged.
- The third time you change the file, and any subsequent times, **sna_node.bk1** is discarded, **sna_node.bk2** is renamed to **sna_node.bk1**, and the existing configuration is saved to **sna_node.bk2**.

This process means that there is a maximum of two backup files for the node configuration file at any time. The same process is used to generate filename extensions for other backup files.

In addition to automatic backups, you should back up configuration files to protect against loss of data under any of the following conditions:

- Before installing a new level of the Linux operating system
- Before installing a new release of CS Linux
- After you create a new configuration

You can back up configuration files using the following commands:

```
cd /etc/opt/ibm/sna
tar -cvf Devicename sna_node.cfg sna.net sna_tps sna_domn.cfg
ibmcs.*
```

Restoring a Backup Copy of CS Linux Configuration Files

To restore CS Linux configuration files that were backed up as described in "Backing Up CS Linux Configuration Files," do the following:

1. Ensure that CS Linux is not active. To determine whether it is, enter the following command:

   ```
   snaadmin status_node
   ```

   If CS Linux is active, the command displays information about the local node's status; otherwise it displays a message indicating that CS Linux is inactive.

   If CS Linux is active, enter the following command to deactivate it:

   ```
   sna stop
   ```

2. Enter the following commands:

   ```
   cd /etc/opt/ibm/sna
   tar -xvf Devicename
   ```

   In this command, *Devicename* is the path and file name of the device you used when backing up the files.

   This command overwrites any existing configuration files with the same names in the **/etc/opt/ibm/sna** directory.

Reinitializing Configuration Files

If CS Linux configuration files are inadvertently modified so that the information in them can no longer be used, you may need to reinitialize the files so that you can reconfigure CS Linux as though it were newly installed. This should be done only if you are sure the configuration information cannot be salvaged.

Note: If you have backup configuration files that are valid, you can copy those files to the **/etc/opt/ibm/sna** directory and use them to initialize the node using the **sna start** command.

You can reinitialize the following configuration files:

- Node configuration file **sna_node.cfg**
- Domain configuration file **sna_domn.cfg**
- TP configuration file **sna_tps**
- SSL keyring database file and password stash file

Perform the following steps to reinitialize configuration files:

1. Exit the administration program if it is active, and disable CS Linux by issuing the following command:

   ```
   sna stop
   ```

2. Back up the existing configuration files by copying any files you are reinitializing to a different location.
3. Delete the files you are reinitializing.
4. If you deleted the domain configuration file, issue the following command to recreate it (by copying from the empty domain configuration file delivered with CS Linux):

   ```
   cp -p /opt/ibm/sna/samples/empty.cfg /etc/opt/ibm/sna/sna_domn.cfg
   ```

   This command creates a new domain configuration file, which is required to start CS Linux.
5. If you deleted the SSL keyring database file, issue the following command to recreate it (by copying from the sample file delivered with CS Linux):

   ```
   cp -p /opt/ibm/sna/samples/ibmcs.* /etc/opt/ibm/sna
   ```

6. Issue the following command to restart CS Linux:

   ```
   sna start
   ```

7. Start the Motif administration program:

   ```
   xsnaadmin &
   ```

   If the **sna_node.cfg** file does not exist, the administration program prompts you to configure the node. You can continue by configuring the node and the other resources as described in Chapter 8, “Configuring and Using CS Linux,” on page 65 or *IBM Communications Server for Data Center Deployment on Linux Administration Guide*.

If you used a valid **sna_node.cfg** file, the new configuration file is used to initialize the node.

Uninstalling CS Linux

You can uninstall the CS Linux product at any time. Use the following procedure:

1. Log in with root privileges.
2. Ensure that CS Linux is not active. To determine whether it is, enter the following command:

   ```
   snaadmin status_node
   ```

   If CS Linux is active, the command displays information about the local node's status; otherwise it displays a message indicating that CS Linux is inactive.

   If CS Linux is active, enter the following command to deactivate it:

   ```
   sna stop
   ```
3. Remove the CS Linux package and associated software packages by using the following instructions:

   ```
   rpm -e ibm-commserver-ptf
   rpm -e ibm-commserver-docs
   rpm -e ibm-commserver-ecl
   rpm -e ibm-commserver
   /sbin/shutdown -r now
   ```

Chapter 4. Installing IBM Remote API Clients on Linux

This chapter describes how to install the IBM Remote API Client on Linux, which enables a Linux workstation to run SNA applications without having a complete SNA stack installation. A Remote API Client on Linux can connect to one or more CS Linux servers (or CS/AIX servers, but not both at the same time) using a TCP/IP network. (CS Linux servers cannot operate in the same domain as CS/AIX servers.)

This chapter applies to IBM Remote API Clients running on 32–bit Intel (i686), 64–bit AMD64/Intel EM64T (x86_64) and pSeries (ppc64) computers. If you are installing the IBM Remote API Client on a System z computer (s390x), refer to Chapter 5, "Installing IBM Remote API Clients on Linux for System z," on page 43.

The installation program and associated files, including the IBM Remote API Client README file, are located on the installation CD, in the appropriate directory for your client type:

Client Type	Directory On CD
32–bit Intel (i686)	**/ibm-commserver-clients/linux**
64–bit AMD64/Intel EM64T (x86_64)	**/ibm-commserver-clients/linux-x86_64**
pSeries (ppc64)	**/ibm-commserver-clients/linux—ppc64**

You are recommended to read the IBM Remote API Client README file before installing the software.

If you are upgrading from an earlier version of CS Linux and the Remote API Clients, you are recommended to upgrade all the servers before upgrading the Remote API Clients. See "Migrating from previous levels of CS Linux" on page 26 for more details.

Hardware and Software Requirements

Hardware Requirements

The IBM Remote API Client requires a computer supported by one of the Linux distributions below.

Use the command **uname -m** to verify the CPU class of your target computer. The following table shows the appropriate hardware for each client type and the response from the **uname -m** for this hardware.

Client Type	Hardware	uname response
32–bit Intel	Pentium II or later 32–bit Intel system, or Opteron-based system	`i686`
64–bit AMD64/Intel EM64T	x86_64 (AMD64 or Intel EM64T) system	`x86_64`
pSeries	pSeries POWER6® system	`ppc64`

Linux Operating System Version

The current version of the IBM Remote API Client has been tested with the following Linux operating system versions. It may also run satisfactorily on other Linux distributions.

- RedHat Enterprise Linux 5 (RHEL5)
- RedHat Enterprise Linux 6 (RHEL6)
- SUSE Linux Enterprise Server 10 (SLES10)
- SUSE Linux Enterprise Server 11 (SLES11)

Refer to the **README** file on the installation CD for details of which optional packages may be required.

Java

If you use the Java CPI-C API, you will require Java software. Refer to the **README** file on the installation CD for details.

GSKIT

If the client will connect to CS Linux servers using HTTPS, you will require GSKIT software to enable HTTPS access to the servers through a WebSphere server. See the **README** file on the installation media for more information about obtaining and installing the GSKIT software.

Displaying Product Installation Details

You can display information about the Remote API Client and related software packages that are already installed. To list all the installed packages, use the following command:

```
rpm -q -a
```

To view more details of a specific package, use the following command:

```
rpm -q -i packagename
```

packagename is the base name of the installed package, for example **ibm-commserver-client**.

Setting environment variables for use by CS Linux

CS Linux uses a number of environment variables to control its operation, which must be set before you enable the CS Linux software on the client. The simplest way to manage these is to set them in a text file that CS Linux reads on startup.

In the CS Linux configuration directory, **/etc/opt/ibm/sna**, create a text file named **environment**, and set each environment variable in a separate line. For example:

```
export LANG=en_US
export PATH=”$PATH:/opt/ibm/sna/bin”
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

If you are running 64-bit applications, change the environment variable as below:

```
export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64
export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64
```

Setting the Language Environment Variable

Use the following command to change the LANG variable to indicate the language you want to use:

```
export LANG=language
```

Replace *language* with the identifier for the language you want to use, which can be one of the following:

Identifier	Language
en_US	English (United States)
ja_JP	Japanese (PC)
de_DE	German
es_ES	Spanish
fr_FR	French
ko_KR	Korean
pt_BR	Portuguese
zh_CN	Chinese (Simplified EUC)
zh_TW	Chinese (Traditional)

Installing the Remote API Client on Linux

After you have installed the pre-requisite software, you are ready to install the IBM Remote API Client.

If you have a previous level of IBM Remote API Client already installed, follow the steps in section “Uninstalling the Remote API Client on Linux” on page 41 to remove it before installing this new level. Any configuration information will be left in place for use by the new installation.

1. Log in with root privileges.
2. Mount the CD and make it the current directory.

   ```
   mount /dev/cdrom
   cd /media/cdrom
   ```

   The directory name **/media/cdrom** may be different if you have a DVD drive. Use the command **df** to see where Linux mounted the CD.
3. Change to the appropriate subdirectory on the CD, and run the shell script to install the client. The example below shows the **/linux** subdirectory for a 32–bit Intel (i686) client; replace this with **/linux-x86_64** or **/linux-ppc64** if required.

   ```
   cd ibm-commserver-clients/linux
   ./installibmcscli
   ```

   The shell script will test for certain pre-requisites and issue warning messages if they are not met. You will be prompted to read and accept the license agreement, then the script will install the RPMs.
4. Create the client network data file to specify the CS Linux servers to which the client can connect. You may also want to set up additional configuration to enable the advanced support for tracking LUs and mapping Local LU aliases; this allows you to integrate the clients more dynamically and manage the LU

resources in a domain of servers. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

5. Add the IBM Remote API Client binary directories to your PATH. You may want to change your profile to do this automatically:

   ```
   export PATH=”$PATH:/opt/ibm/sna/bin”
   export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
   export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
   ```

   If you are running 64-bit applications, change the environment variable as below:

   ```
   export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64
   export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64
   ```

   For Java CPI-C applications you should also set the following environment variable:

   ```
   export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar
   ```

   For some applications you may also need to set the `LD_PRELOAD` environment variable, but you should not make this a global change in your profile:

   ```
   export LD_PRELOAD=/usr/lib/libpLiS.so
   ```

6. Start the IBM Remote API Client. After installation this will happen automatically when the machine is rebooted. Make sure you are not still in the CD's directories when you do this.

   ```
   cd /
   sna start
   ```

Note: Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. See “Setting up HTTPS security certificates using GSKIT” for more information.

You will also need to update the client network data file to specify the name of the WebSphere server that provides HTTPS support. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

Setting up HTTPS security certificates using GSKIT

If the client will connect to CS Linux servers using HTTPS, it must have the GSKIT key manager software installed. See the **README** file on the installation media for more information about obtaining and installing the GSKIT software.

Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. Take the following steps.

1. Run the GSKIT key manager using the following command:

   ```
   /opt/ibm/sna/bin/snakeyman
   ```

   From within the key manager user interface, open the key database file **/etc/opt/ibm/sna/ibmcs.kdb**, which is in CMS format.

2. The initial password for the key database is `ibmcs`. Before setting up the security certificates, you **must** change this password to keep your configuration secure. In the dialog for changing the password, you will need to mark the

checkbox 'Stash the password to a file?' to ensure that the new password is saved so that the client can open the key database.

3. Obtain a copy of the Certificate Authority (CA) certificate that was used to sign the Web Server's security certificate, and install it in the key database. To do this, select Signer Certificates from the key manager user interface and click on Add.
4. If the WebSphere server is configured to require client security certificates, the client must have a certificate issued by a CA whose own certificate is in the Web Server's security certificate database. To request a new certificate:
   a. Select Create, New Certificate Request from the key manager user interface, and fill in the requested details.
   b. Save the certificate, extract it to a file and send it to the CA.
   c. When the certificate is issued, store it in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.

   As a temporary measure for your own internal testing, you can create a self-signed client certificate rather than obtaining a certificate from the CA. However, this does not provide the required level of security and must not be used in a live system. To create a self-signed certificate:
   a. Select Create, New Self-Signed Certificate from the key manager user interface, and fill in the requested details.
   b. Save the certificate and extract it to a file.
   c. Store the certificate file in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.
5. Exit the GSKIT key manager when you have finished configuring certificates.

Uninstalling the Remote API Client on Linux

You can uninstall the Remote API Client on Linux by using the following commands.

```
/opt/ibm/sna/bin/sna stop
rpm -e ibm-commserver-ptf
rpm -e ibm-commserver-docs
rpm -e ibm-commserver-ecl
rpm -e ibm-commserver-cli
rpm -e ibm-commserver
rpm -e gskssl32 gskcrypt32
/sbin/shutdown -r now
```

Not all of the packages listed in these commands will be installed on every system.

Uninstalling IBM Remote API Client on Linux will leave any customized configuration information behind for use by a later installation.

Chapter 5. Installing IBM Remote API Clients on Linux for System z

This chapter describes how to install the IBM Remote API Client on Linux for System z, which enables a System z mainframe to run SNA applications without having a complete SNA stack installation. A Remote API Client on Linux for System z can connect to one or more CS Linux servers (or CS/AIX servers) using a TCP/IP network.

You are recommended to read the IBM Remote API Client README file before installing the software. This file is located in the **/ibm-commserver-clients/linux-systemz** directory on the installation CD.

If you are upgrading from an earlier version of CS Linux and the Remote API Clients, you are recommended to upgrade all the servers before upgrading the Remote API Clients. See "Migrating from previous levels of CS Linux" on page 26 for more details.

Hardware and Software Requirements

Hardware Requirements

The IBM Remote API Client requires a 64–bit System z system supported by one of the Linux distributions listed in "Linux Operating System Version."

Use the command **uname -m** to verify the CPU class. It must report `s390x` to indicate a 64–bit environment.

Linux Operating System Version

The current version of the IBM Remote API Client has been tested with the following Linux operating system versions. It may also run satisfactorily on other Linux distributions.

- RedHat Enterprise Linux 5 for System z (RHEL5–s390x)
- RedHat Enterprise Linux 6 for System z (RHEL6–s390x)
- SUSE Linux Enterprise Server 10 for IBM Mainframe (SLES10–s390x)
- SUSE Linux Enterprise Server 11 for IBM Mainframe (SLES11–s390x)

Refer to the **README** file on the installation CD for details of which optional packages may be required.

Java

If you use the Java CPI-C API, you will require Java software. Refer to the **README** file on the installation CD for details.

GSKIT

If the client will connect to CS Linux servers using HTTPS, you will require GSKIT software to enable HTTPS access to the servers through a WebSphere server. See the **README** file on the installation media for more information about obtaining and installing the GSKIT software.

Displaying Product Installation Details

You can display information about the Remote API Client and related software packages that are already installed. To list all the installed packages, use the following command:

```
rpm -q -a
```

To view more details of a specific package, use the following command:

```
rpm -q -i packagename
```

packagename is the base name of the installed package, for example **ibm-commserver-client**.

Setting environment variables for use by CS Linux

CS Linux uses a number of environment variables to control its operation, which must be set before you enable the CS Linux software on the client. The simplest way to manage these is to set them in a text file that CS Linux reads on startup.

In the CS Linux configuration directory, **/etc/opt/ibm/sna**, create a text file named **environment**, and set each environment variable in a separate line. For example:

```
export LANG=en_US
export PATH="$PATH:/opt/ibm/sna/bin"
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

If you are running 64-bit applications, change the environment variables as below:

```
export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64
export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64
```

Setting the Language Environment Variable

Use the following command to change the LANG variable to indicate the language you want to use:

```
export LANG=language
```

Replace *language* with the identifier for the language you want to use, which can be one of the following:

Identifier	Language
en_US	English (United States)
ja_JP	Japanese (PC)
de_DE	German
es_ES	Spanish
fr_FR	French
ko_KR	Korean
pt_BR	Portuguese
zh_CN	Chinese (Simplified EUC)
zh_TW	Chinese (Traditional)

Installing the Remote API Client on Linux for System z

After you have installed the pre-requisite software, you are ready to install the IBM Remote API Client.

If you have a previous level of IBM Remote API Client already installed, follow the steps in section "Uninstalling the Remote API Client on Linux for System z" on page 47 to remove it before installing this new level. Any configuration information will be left in place for use by the new installation.

1. Copy or FTP the **ibm-commserver-client–7.0.0.0–s390x.tgz** file from the **/ibm-commserver-clients/linux-systemz** directory on the CD-ROM to the Linux System z system. Ensure that you use binary mode to copy or FTP the file.
2. Log into the Linux System z system as root.
3. Uncompress and unpack the tar file into an empty temporary directory:

   mkdir /tmp/ibmcs

   cd /tmp/ibmcs

   tar —xzf ibm-commserver-client–7.0.0.0–s390x.tgz
4. Run the **installibmcscli** shell script:

   ./installibmcscli

   This shell script tests for certain pre-requisites and issues warning messages if they are not met. It also prompts you to confirm that you have read and accepted the CS Linux license terms. You can override this prompt by specifying additional parameters on the **installibmcscli** command as described below. When you have responded to the prompt, the shell script installs the **rpm**
5. Create the client network data file to specify the CS Linux servers to which the client can connect. You may also want to set up additional configuration to enable the advanced support for tracking LUs and mapping Local LU aliases; this allows you to integrate the clients more dynamically and manage the LU resources in a domain of servers. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.
6. Add the IBM Remote API Client binary directories to your PATH. You may want to change your profile to do this automatically:

   export PATH="$PATH:/opt/ibm/sna/bin"

   export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib

   export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib

   If you will be running 64–bit applications, change your profile as below:

   export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64

   export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64

   For Java CPI-C applications you should also set the following environment variable:

   export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar

   For some applications you may also need to set the `LD_PRELOAD`environment variable, but you should not make this a global change in your profile:

   export LD_PRELOAD=/usr/lib/libpLiS.so
7. Start the IBM Remote API Client. After installation this will happen automatically when the machine is rebooted. Make sure you are not still in the CD's directories when you do this.

cd /
sna start

8. When you are done with the installation, you can erase the **tgz** file and temporary directory that were created during the installation process.

Note: Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. See “Setting up HTTPS security certificates using GSKIT” for more information.

You will also need to update the client network data file to specify the name of the WebSphere server that provides HTTPS support. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

Setting up HTTPS security certificates using GSKIT

If the client will connect to CS Linux servers using HTTPS, it must have the GSKIT key manager software installed. See the **README** file on the installation media for more information about obtaining and installing the GSKIT software.

Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. Take the following steps.

1. Run the GSKIT key manager using the following command:

   /opt/ibm/sna/bin/snakeyman

   From within the key manager user interface, open the key database file **/etc/opt/ibm/sna/ibmcs.kdb**, which is in CMS format.
2. The initial password for the key database is `ibmcs`. Before setting up the security certificates, you **must** change this password to keep your configuration secure. In the dialog for changing the password, you will need to mark the checkbox 'Stash the password to a file?' to ensure that the new password is saved so that the client can open the key database.
3. Obtain a copy of the Certificate Authority (CA) certificate that was used to sign the Web Server's security certificate, and install it in the key database. To do this, select Signer Certificates from the key manager user interface and click on Add.
4. If the WebSphere server is configured to require client security certificates, the client must have a certificate issued by a CA whose own certificate is in the Web Server's security certificate database. To request a new certificate:
   a. Select Create, New Certificate Request from the key manager user interface, and fill in the requested details.
   b. Save the certificate, extract it to a file and send it to the CA.
   c. When the certificate is issued, store it in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.

   As a temporary measure for your own internal testing, you can create a self-signed client certificate rather than obtaining a certificate from the CA. However, this does not provide the required level of security and must not be used in a live system. To create a self-signed certificate:
   a. Select Create, New Self-Signed Certificate from the key manager user interface, and fill in the requested details.

  b. Save the certificate and extract it to a file.

  c. Store the certificate file in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.

5. Exit the GSKIT key manager when you have finished configuring certificates.

Uninstalling the Remote API Client on Linux for System z

You can uninstall the Remote API Client on Linux for System z by using the following commands.

```
/opt/ibm/sna/bin/sna stop
rpm -e ibm-commserver-ptf
rpm -e ibm-commserver-docs
rpm -e ibm-commserver-ecl
rpm -e ibm-commserver-cli
rpm -e ibm-commserver
rpm -e gskssl32 gskcrypt32
/sbin/shutdown -r now
```

Not all of the packages listed in these commands will be installed on every system.

Uninstalling IBM Remote API Client on Linux for System z will leave any customized configuration information behind for use by a later installation.

Chapter 6. Installing IBM Remote API Clients on AIX Systems

This chapter describes how to install the IBM Remote API Client on AIX, which enables an AIX workstation to run SNA applications without having a complete SNA stack installation. A Remote API Client on AIX can connect to one or more CS Linux servers (or CS/AIX servers) using a TCP/IP network.

You are recommended to read the IBM Remote API Client README file before installing the software. This file is located in the **/ibm-commserver-clients/aix** directory on the installation CD.If you are upgrading from an earlier version of CS Linux and the Remote API Clients, you are recommended to upgrade all the servers before upgrading the Remote API Clients. See "Migrating from previous levels of CS Linux" on page 26 for more details.

Hardware and Software Requirements

Hardware Requirements

The IBM Remote API Client requires a pSeries system supported by one of the AIX operating systems listed in "Operating System Version."

Operating System Version

The current version of the IBM Remote API Client has been tested with the following operating system versions.

- AIX 6.1 or later
- AIX 7.1 or later

  The client can run in the Global Environment, or in a system or application WPAR. You must ensure that any WPAR in which the client runs has a unique hostname that DNS can resolve.

Java

If you use the Java CPI-C API, you will require Java software. The latest Java SDK available from http://www.ibm.com/developerworks/java/jdk satisfies all the requirements.

Install the Java SDK package with the **installp** command.

GSKIT

If the client will connect to CS Linux servers using HTTPS, you will require GSKIT software to enable HTTPS access to the servers through a WebSphere server. See the **README** file on the installation media for more information about obtaining and installing the GSKIT software.

Setting environment variables for use by CS/AIX

CS/AIX uses a number of environment variables to control its operation, which must be set before you enable the CS/AIX software on the client. The simplest way to manage these is to set them in a text file that CS/AIX reads on startup.

In the CS/AIX configuration directory, **/etc/sna**, create a text file named **environment**, and set each environment variable in a separate line. For example:

```
export LANG=en_US
export PATH="$PATH:/usr/bin"
export LD_LIBRARY_PATH=/usr/lib:/usr/lib/sna
export LD_RUN_PATH=/usr/lib:/usr/lib/sna
```

Changing the Language Environment Variable

When you use the Remote API Client, make sure that the `LANG` variable is not set to `C`.

Use the following procedure to show which `LANG` variable is in use or to change the `LANG` variable:

1. From the main SMIT menu, select **System Environments**.
2. From the next SMIT menu, select **Manage Language Environment**.
3. From the next SMIT menu, select **Change/Show Primary Language Environment**.
4. From the next SMIT menu, select **Change/Show Cultural Convention, Language, or Keyboard**.
5. Select the language you want to use. For example, if you are using U.S. English messages, select `en_US`.

Installing the Remote API Client on AIX

After you have installed the pre-requisite software, you are ready to install the IBM Remote API Client.

If you have a previous level of IBM Remote API Client already installed, follow the steps in section "Uninstalling the Remote API Client on AIX" on page 52 to remove it before installing this new level. Any configuration information will be left in place for use by the new installation.

Installing the Remote API Client by copying files to your AIX workstation

To install the Remote API Client, take the following steps.

1. Copy or FTP the **sna.client.7.0.0.0.I** file from the **/ibm-commserver-clients/aix** directory on the CD-ROM to the AIX workstation. Ensure that you use binary mode to copy or FTP the file.
2. Log into the AIX workstation as root.
3. Install the AIX Client using either **smit** or **installp**. For instructions on how to do this, see the **README** file in the **/ibm-commserver-clients/aix** directory on the installation CD.
4. When the installation process has completed, you can delete the **sna.client.7.0.0.0.I** file from the working directory.
5. Create the client network data file to specify the CS Linux servers to which the client can connect. You may also want to set up additional configuration to enable the advanced support for tracking LUs and mapping Local LU aliases; this allows you to integrate the clients more dynamically and manage the LU resources in a domain of servers. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.
6. Start the IBM Remote API Client. After installation this will happen automatically when the machine is rebooted.

cd /
sna start

Note: Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. See "Setting up HTTPS security certificates using GSKIT" for more information.

You will also need to update the client network data file to specify the CS Linux servers to which the client can connect and the name of the WebSphere server that provides HTTPS support. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

Installing the Remote API Client from the CD

To install the Remote API Client, take the following steps.

1. Log into the AIX workstation as root.
2. Mount the CD on the AIX workstation, using the following command.

   mount -o ro /dev/cd0 /mnt
3. Install the AIX Client using either **smit** or **installp**. For instructions on how to do this, see the **README** file in the **/ibm-commserver-clients/aix** directory on the installation CD.
4. When the installation process has completed, unmount the CD using the following command.

   unmount /mnt
5. Create the client network data file to specify the CS Linux servers to which the client can connect. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.
6. Start the IBM Remote API Client. After installation this will happen automatically when the machine is rebooted. Make sure you are not still in the CD's directories when you do this.

   cd /
   sna start

Note: Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. See "Setting up HTTPS security certificates using GSKIT" for more information.

You will also need to update the client network data file to specify the CS Linux servers to which the client can connect and the name of the WebSphere server that provides HTTPS support. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

Setting up HTTPS security certificates using GSKIT

Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. Take the following steps.

1. Run the GSKIT key manager using the following command:

/usr/bin/snakeyman

From within the key manager user interface, open the key database file **/etc/sna/ibmcs.kdb**, which is in CMS format.

2. The initial password for the key database is `ibmcs`. Before setting up the security certificates, you **must** change this password to keep your configuration secure. In the dialog for changing the password, you will need to mark the checkbox 'Stash the password to a file?' to ensure that the new password is saved so that the client can open the key database.
3. Obtain a copy of the Certificate Authority (CA) certificate that was used to sign the Web Server's security certificate, and install it in the key database. To do this, select Signer Certificates from the key manager user interface and click on Add.
4. If the WebSphere server is configured to require client security certificates, the client must have a certificate issued by a CA whose own certificate is in the Web Server's security certificate database. To request a new certificate:
   a. Select Create, New Certificate Request from the key manager user interface, and fill in the requested details.
   b. Save the certificate, extract it to a file and send it to the CA.
   c. When the certificate is issued, store it in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.

   As a temporary measure for your own internal testing, you can create a self-signed client certificate rather than obtaining a certificate from the CA. However, this does not provide the required level of security and must not be used in a live system. To create a self-signed certificate:
   a. Select Create, New Self-Signed Certificate from the key manager user interface, and fill in the requested details.
   b. Save the certificate and extract it to a file.
   c. Store the certificate file in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.
5. Exit the GSKIT key manager when you have finished configuring certificates.

Uninstalling the Remote API Client on AIX

You can uninstall the Remote API Client by using the following commands.

1. Stop the client software if it is running, using the following command.

   sna stop
2. Log in with root privileges.
3. Remove the Remote API Client package and associated software packages by using one of the following commands.

   To remove the package using **installp**:

   installp —u sna.client

   To remove the package using **smit**:

   smit remove

Chapter 7. Planning for and Installing the Remote API Client on Windows

This chapter describes how to install the IBM Remote API Client on Windows, which enables a PC to run SNA applications without having a complete SNA stack installation on the PC. A Remote API Client on Windows can connect to one or more CS Linux servers (or CS/AIX servers) using a TCP/IP network.

If you are upgrading from an earlier version of CS Linux and the Remote API Clients, you are recommended to upgrade all the servers before upgrading the Remote API Clients. See "Migrating from previous levels of CS Linux" on page 26 for more details.

There are two variants of the IBM Remote API Client on Windows, depending on the specific hardware and Windows version you are using. The information in this chapter applies to both variants except where differences are noted explicitly.

- The 32–bit client runs on a 32–bit Intel-based computer running Windows 2003, Windows XP, 32–bit Windows Vista, 32–bit Windows Server 2008, 32–bit Windows 7, or32–bit Windows 8.
- The x64 client runs on an AMD64 or Intel EM64T computer running Microsoft Windows 2003 Server x64 Edition, Windows XP Professional x64 Edition, 64–bit Windows Vista, 64–bit Windows Server 2008, 64–bit Windows 7, or 64–bit Windows 8.

The interfaces provided by the IBM Remote API Client on Windows are broadly compatible with those provided by the IBM Communications Server for Windows and Microsoft Host Integration Server products.

The IBM Remote API Client on Windows Software Development Kit (SDK) is an optional package that allows you to use the Remote API Client to develop application programs using the APPC, CPI-C, LUA, and CSV APIs. Refer to the appropriate programmer's reference guide for more information about these APIs. You do not need to install this package if the Remote API Client will be used only to run existing applications (not to develop new ones).

Hardware and Software Requirements

To run the **Setup** program and the Remote API Client on Windows, the computer must meet the following requirements:

- It must be running one of the following operating systems. For up-to-date information about specific version numbers that are supported for each operating system version, and any additional requirements for specific versions, please refer to the Windows client information in the **README** file on the installation CD for more details.
  - For the 32–bit Windows client:
    - Windows XP
    - Windows 2003
    - 32–bit Windows Vista
    - 32–bit Windows Server 2008
    - 32–bit Windows 7

    - 32–bit Windows 8
  - For the x64 Windows client:
    - Microsoft Windows XP Professional x64 Edition
    - Microsoft Windows 2003 Server x64 Edition
    - 64–bit Windows Vista
    - 64–bit Windows Server 2008
    - 64–bit Windows 7
    - 64–bit Windows 8
- It must have access to one or more CS Linux servers using one of the following mechanisms:
  - Access to the server over a TCP/IP network
  - Access to a WebSphere server that provides HTTPS access to CS Linux servers.

Note: Depending on the Windows version that you are using, or the specific Remote API Client features tha you want to use, there may be some additional configuration that you need to perform before you can install and use the Remote API Client on Windows. Refer to the Windows client information in the **README** file on the installation CD for more details. In particular, if the client accesses its servers using HTTPS, you need to install the GSKIT software.

Accessing the Setup Program

The Remote API Client and SDK software and the **Setup** program are included on the installation CD in Windows format, so that you can install them from the CD on the Windows computer. You must install the Remote API Client software on each Windows client PC. The SDK is required only if you will be using the client to develop new applications using the Windows Remote APIs, and is not required if you will be using it only to run existing applications.

The Remote API Client on Windows installation image is a self-extracting ZIP executable file, delivered on the installation CD.

- For the 32–bit client, it is **i_w32cli.exe** in the directory **/ibm-commserver-clients/windows** on the CD.
- For the x64 client, it is **i_w64cli.exe** in the directory **/ibm-commserver-clients/win-x64** on the CD.

You can copy this file to other Windows PCs across the network, so that you can install them without direct access to the CS Linux delivery CD. When you run this executable, it unzips the installation image and automatically runs the **Setup** program. If you simply want to unzip the installation image to a temporary directory, for example to run the **Setup** program from the command line, you can do so by loading the self-extracting ZIP executable into your unzip program.

The first time you run the **Setup** program on a particular computer, the program runs from the selected source. The program handles the complete installation process, sets up a basic configuration, and also installs and creates an icon for itself. After installation is complete, you can use the **Setup** program (either by selecting it from the File Manager or by selecting its icon) if you need to reinstall the software.

After you have extracted the Remote API Client installation image into a temporary directory, you can install the software in either of two ways:

- Run the **Setup** program through Windows, as explained in "Installing Remote API Client on Windows Using the Setup Program." You must use this method if you want to install the SDK.
- Enter the **setup** command from the command line, as explained in "Installing Remote API Client Software from the Command Line" on page 58. This method does not allow you to install the SDK.

Note: Before the IBM Remote API Client can connect to servers using HTTPS, you need to update the client network data file to specify the CS Linux servers to which the client can connect and the name of the WebSphere server that provides HTTPS support. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

Installing Remote API Client on Windows Using the Setup Program

Run the **setup** program, either automatically as part of executing the self-extracting ZIP executable **i_w32cli.exe** (32–bit client) or **i_w64cli.exe** (x64 client), or manually from the command line. The program first displays a Choose Setup Language screen.

1. Select the language that you want to use for installing and configuring the Remote API Client, and choose **OK**.

   The program displays a Welcome screen that introduces you to the **Setup** program.
2. Choose **Next** to continue with the installation.

   The program displays the Software Licensing Agreement, which you should read and understand.
3. If you are happy to accept the licensing terms, choose **Accept** to continue.

   The program prompts you to specify a destination directory into which the files are to be installed.
4. Enter the destination directory.

   The program asks you to choose the type of installation you want:

   Standard
   Choose this option if you do not need to install the SDK. The SDK is required only if you will be using the client to develop new applications using the Windows Remote APIs, and is not required if you will be using it only to run existing applications.

   Developer
   Choose this option if you need to install the SDK: that is, if you will be using the client to develop new applications using the Windows Remote APIs.

   Note: If you want to install the SDK, you must choose **Developer**.
5. Choose the installation type.

   The program then asks you to enter the name of the program folder in which you wish icons for the Remote API Client on Windows to appear.
6. Enter the folder name.

7. If the System directory already contains **.DLL** files with names that are the same as files used by this **Setup** program but are not Remote API Client files (for example, files from some other SNA software), the program prompts you to do one of the following:
   - Copy the Remote API Client **.DLL** files over the existing **.DLL** files
   - Copy the existing **.DLL** files to a subdirectory named **OTHERSNA** within the installation directory, and then install the Remote API Client **.DLL** files. This option enables you to restore the original setup from before the Remote API Client installation if you uninstall the files at a later time (see "Uninstalling the Remote API Client Software" on page 63).
   - Cancel the client software installation.

   If the Remote API Client **.DLL** files are already present, the **Setup** program displays a message indicating this. New **.DLL** files will overwrite the existing **.DLL** files only if the existing files have lower version numbers than the **Setup** program **.DLL** files.
8. At this point, the **Setup** program copies files from the specified source, and installs them in the appropriate places. During this process, an information bar displays what portion of the installation is complete. The **.DLL** files are copied into the System or equivalent directory, and the other files are copied into the destination directory you specified in Step 2. During each file transfer operation, a record is written to the **setup.log** file, which is created in the directory you specified. If any of the files to be written over is "read only", or any file cannot be copied for any other reason, the new files are removed and you receive a message advising you to look at the **setup.log** file.
9. If the source from which you are running the **Setup** program does not contain all the required files, the program prompts you for a directory name. Enter the name of a directory in which the required files are located.

   If the information you specified is not sufficient to locate copies of the Remote API Client files, the program displays this screen again.
10. When the required files have been copied, the **Setup** program displays the Configuration window.

    Default configuration values are taken from the domain configuration file. For more information, see the *IBM Communications Server for Data Center Deployment on Linux Administration Guide*. If you do not want to use these default values, you can configure them as shown below:

    Domain

    Specify the CS Linux client/server domain name.

    If the client uses IPv6 addressing, you must configure the following settings. They are optional if the client uses IPv4 addressing.

    Server Name

    The screen shows a list of up to nine servers to which this client can connect. The order in which servers appear in this list is the order in which the client selects these servers. If the client cannot connect to the first server on the list, the next server is tried.

    Specify each server name in one of the following formats:
    - If the client uses HTTPS to access its servers, specify the name of the WebSphere server that provides HTTPS support and the name of the CS Linux server, in the following format:

      webservername **:** *servername1*

      This assumes that WebSphere is set up to use the default port 443 for HTTPS connections. If your network administrator has configured WebSphere to use a different port number, include the port number in the following format:

webservername **:** *portnumber* **:** *servername1*

For more details about configuring WebSphere to support HTTPS connections, refer to "Configuring WebSphere Application Server" on page 31.

- If the client does not use HTTPS to access its servers, specify the name of the first server it should try to contact. The *webservername* and *portnumber* parameters are not used and should not be specified.
- If the client is on the same private network as its servers, you can specify that it should find a server running CS Linux by using a UDP broadcast message to all computers on its TCP/IP subnet (or on all subnets that it can access, if the client computer contains more than one LAN adapter card). To do this, specify * (an asterisk character) instead of the first server name.

  This option is available only if the client uses IPv4 addressing. UDP broadcasts are not supported for IPv6.

  The client retries the broadcast every 10 seconds, up to the number of attempts specified by the *broadcast_attempt_count* parameter, until it contacts a server. If the limit specified by *broadcast_attempt_count* is reached before a server has been contacted, the client then tries using directed messages to one or more named servers (specified by the following lines of the file).

Note: If you are not using UDP broadcasts, you must specify the names of all servers that this client needs to access. The client can use resources on the servers that are specified in this file, but cannot use resources on other servers.

The following settings are optional:

Advanced

To supply additional values in place of the defaults supplied by the **Setup** program, click on the **Advanced** button at the bottom of the window. The **Setup** program displays the Advanced Options window, which contains advanced settings for Windows client configuration. Most users can use the default settings for these parameters, so you probably do not need to alter the settings in that dialog.

For more information about these parameters, see "Advanced Options for Remote API Client Configuration" on page 58.

For more information about any of the configuration parameters or settings, click on **Help**.

11. When you have completed the Configuration window, click on **OK**. The **Setup** program displays a message if you have not completed this screen properly.
12. When the installation has successfully completed, the Finish window is displayed. You can select either or both of the following actions to be taken after you exit the installation program:

View README file

View the **README** file.

Start client

Begin running this CS Linux client.

Choose **Finish** to exit the installation program.

Advanced Options for Remote API Client Configuration

The Advanced Options window enables you to configure some advanced parameters for the Remote API Client. Most users do not need to alter these parameters, but you can adjust the default settings if necessary.

LAN access time-out
: Specify the time in seconds that the client's connection to a server can remain idle before it is closed. When this check box is empty, no LAN access time-out has been specified (and so an infinite time-out will be used). If you check this box, you can enter a time-out value in seconds in the adjacent field. The minimum value is `60` (for 60 seconds); if you leave the box blank or specify a value lower than `60`, the Remote API Client uses the minimum value `60`.

Max. broadcast attempts
: Specify the maximum number of times the client attempts to connect to a server by broadcast. When the Advanced Options window is opened, the default value of `5` is displayed. The value in this box is used only if the client uses UDP broadcasts: this is indicated by `*` (an asterisk character) instead of the first server name.

Reconnect time-out
: Specify the time in seconds that the client waits before attempting to reconnect to a server after the server has gone down. When the Advanced Options window is opened, the default value of `200` is displayed.

For more information about these parameters, press **Help**.

When you have completed the Advanced Options window, click on **OK**. If you have completed the screen properly, the **Setup** program returns to the Configuration window. If you are installing a new Remote API Client, return to Step 11 on page 57. Otherwise, click on the **OK** button in the Configuration dialog to complete the configuration.

Installing Remote API Client Software from the Command Line

Note: If you want to install the SDK, you must use the **Setup** program, as explained in “Installing Remote API Client on Windows Using the Setup Program” on page 55. You cannot install the SDK from the command line.

After you have extracted the Remote API Client installation image into a temporary directory, you can install the Remote API Client software from the command line instead of using the **Setup** program through Windows. At the command line, enter the **setup** command with one or more options. You can enter these options in uppercase or lowercase, and can precede them with a `/` (slash) or `-` (hyphen). If a parameter, such as *folder*, is a string that contains a space, you must enclose the string inside double quotes.

After you enter the **setup** command, the **Setup** program prompts you for any information you have not included on the command line, and displays confirmation messages at various stages of the setup. If you do not want the **Setup** program to prompt you, use the **-accept -s** option to run the program in silent mode, accepting the terms of the Software License Agreement.

Following are the **setup** command options:

-accept -s
Run the installation in silent mode, accepting the terms of the Software License Agreement. This agreement can be found in the **license** subdirectory of the Windows installation image.

The **-s** option must be the last one in the command line, and you must be sure you have specified the domain name (using the **-i** option) and any other parameters that you want to specify. When the installation runs in silent mode, it does not prompt you for any parameters or display confirmation messages. Any command-line arguments after **-s** are ignored.

-f2
Specify the full pathname of the installation log file that is created during silent mode installation (using the **-s** option).

If you do not specify this option, the file is created as **setup.log** in the directory from which you run the installation program. If you are installing in silent mode from the CD drive, you must specify this option to ensure that the file is created on your computer (because it cannot be created on the CD drive).

-k*folder*
Specify the Program folder.

-l*languagecode*
Specify the language variant of the client that you want to use. If you do not specify this option, the default is to install the English variant.

Use one of the following codes (including the leading `0x`) to indicate the language you want. For example, use **-l0x000c** to install the French variant of the client.

English (default):	`0x0009`
Chinese:	`0x0804`
French:	`0x000c`
German:	`0x0007`
Japanese:	`0x0011`
Korean:	`0x0012`
Portuguese:	`0x0016`
Spanish:	`0x000a`
Taiwanese:	`0x0404`

-p*directory*
Specify the install directory.

-i*domain*
Specify a domain name for this client. This parameter is required; there is no default.

-w*directory*
Specify the source directory containing CS Linux client software files if the source is located on a disk or CD. Otherwise, use the **-v** option.

-v*server*
Specify the server from which the client software files are to be downloaded. You can specify either the server name or TCP/IP address. If you are copying the source files from a disk or CD, use the **-w** option instead of the **-v** option.

-u*server*
Specify the name of a server to be included in the list of servers this client can access, in one of the following formats:

- If the client uses HTTPS to access its servers, specify the name of the WebSphere server that provides HTTPS support and the name of the CS Linux server, in the following format:

  webservername **:** *servername1*

  This assumes that WebSphere is set up to use the default port 443 for HTTPS connections. If your network administrator has configured WebSphere to use a different port number, include the port number in the following format:

  webservername **:** *portnumber* **:** *servername1*

  For more details about configuring WebSphere to support HTTPS connections, refer to "Configuring WebSphere Application Server" on page 31.
- If the client does not use HTTPS to access its servers, specify the name of the server it should try to contact.
- If the client is on the same private network as its servers, you can specify that it should find a server running CS Linux by using a UDP broadcast message to all computers on its TCP/IP subnet (or on all subnets that it can access, if the client computer contains more than one LAN adapter card). To do this, specify `*` (an asterisk character) instead of the first server name.

  This option is available only if the client uses IPv4 addressing. UDP broadcasts are not supported for IPv6.

  The client retries the broadcast every 10 seconds, up to the number of attempts specified by the *broadcast_attempt_count* parameter, until it contacts a server. If the limit specified by *broadcast_attempt_count* is reached before a server has been contacted, the client then tries using directed messages to one or more named servers (specified by the following lines of the file).

Note: If you are not using UDP broadcasts, you must specify the names of all servers that this client needs to access. The client can use resources on the servers that are specified in this file, but cannot use resources on other servers.

-o Overwrite existing **.DLL** files. If the Remote API Client **.DLL** files are already present, the **Setup** program overwrites these files even if they have a higher version number than the **Setup** program **.DLL** files.

-y Save existing **.DLL** files. If the Remote API Client **.DLL** files already exist in the required directories, the **Setup** program copies the existing **.DLL** files to a subdirectory of the install directory, and then installs the Remote API Client **.DLL** files. The copies in the subdirectory ensure that if you uninstall the Remote API Client software, the uninstall process will be complete.

-n Cancel the installation if existing Remote API Client **.DLL** files are found.

-a*timeout*

Specify the LAN access time-out in seconds. This is the length of time the client's connection to a server can remain idle before being closed. The value `0` indicates no time-out.

-b*max-broadcast*

This option is used only if the client uses UDP broadcasts: this is indicated by `*` (an asterisk character) instead of the first server name.

Specify the maximum number of UDP broadcast attempts. A UDP broadcast is a client's attempt to connect to any server in the domain rather than to a specific server. The value `0` indicates no broadcast attempts are made.

-j*reconnect-timeout*
: Specify the time in seconds that the client waits before attempting to reconnect to a server after the server has gone down.

To view summary help information about these options on the screen, use the command **setup -h -l***languagecode*, where *languagecode* is the code (as described above) for the language in which you want to view the information. For example, use the command **setup -h -l0x0009** to view the help information in English. You can also replace **-h** with **-?**, for example **setup -? -l0x000a** to view the help information in Spanish.

The following is an example command line to install the Remote API Client:

setup -imy_domain -userver1.company.com -userver2.company.com -b0 -j30 -accept -s -f2C:\instrapi.log -y

In this example:

- The client is installed in the domain **my_domain**.
- The client has access to two servers in the same private network as the client, and does not use UDP broadcasts to contact any other servers. It waits for 30 seconds before reconnecting if it loses contact with a server.
- The installation runs in silent mode, writing its installation log information to the file **C:\instrapi.log** on the client.
- Existing copies of Remote API Client **.DLL** files are saved to a subdirectory before the new files are installed.
- No language option is specified, so the default (English) is used.

Setting up HTTPS security certificates using GSKIT

Before the IBM Remote API Client can connect to servers using HTTPS, you need to use the GSKIT key manager program to set up the security certificate configuration on the client. Take the following steps.

1. Run the GSKIT key manager program, which is *installdir***\snakeyman.exe**. *installdir* represents the directory in which you installed the client software, which is **C:\IBMCS\w32cli** (32–bit client) or **C:\IBMCS\w64cli** (64–bit client) unless you specified a different location during the client installation.

   From within the key manager user interface, open the key database file *installdir***\ibmcs.kdb**, which is in CMS format.
2. The initial password for the key database is `ibmcs`. Before setting up the security certificates, you **must** change this password to keep your configuration secure. In the dialog for changing the password, you will need to mark the checkbox 'Stash the password to a file?' to ensure that the new password is saved so that the client can open the key database.
3. Obtain a copy of the Certificate Authority (CA) certificate that was used to sign the Web Server's security certificate, and install it in the key database. To do this, select Signer Certificates from the key manager user interface and click on Add.

4. If the WebSphere server is configured to require client security certificates, the client must have a certificate issued by a CA whose own certificate is in the Web Server's security certificate database. To request a new certificate:
   a. Select Create, New Certificate Request from the key manager user interface, and fill in the requested details.
   b. Save the certificate, extract it to a file and send it to the CA.
   c. When the certificate is issued, store it in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.

   As a temporary measure for your own internal testing, you can create a self-signed client certificate rather than obtaining a certificate from the CA. However, this does not provide the required level of security and must not be used in a live system. To create a self-signed certificate:
   a. Select Create, New Self-Signed Certificate from the key manager user interface, and fill in the requested details.
   b. Save the certificate and extract it to a file.
   c. Store the certificate file in the Web Server's database. To do this, select Personal Certificates from the key manager user interface and click on Receive.
5. Exit the GSKIT key manager when you have finished configuring certificates.

Customizing the Remote API Client Software after Installation

You can change any of the customized settings any time after the initial installation by running the **Configuration Utility** program, located in the IBM Remote API program group. The program displays the same Configuration window that was displayed in the initial install process. You can change the information in any field by following the procedure in "Installing Remote API Client on Windows Using the Setup Program" on page 55.

If you did not install the SDK files during the initial installation and you now want to add them, you can do this by running the Setup program again and choosing **Developer** for the installation type.

You may also want to set up additional configuration to enable the advanced support for tracking LUs and mapping Local LU aliases; this allows you to integrate the clients more dynamically and manage the LU resources in a domain of servers. See the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux Administration Guide* for more details.

Reinstalling the Remote API Client Software

You can reinstall the Remote API Client software at any time, for example when you wish to upgrade the software.

To do this, run the Setup program as before, using the instructions in "Installing Remote API Client on Windows Using the Setup Program" on page 55 or "Installing Remote API Client Software from the Command Line" on page 58. The **Setup** program displays the location from which the client software files were copied during the initial installation. Click on **OK** to get new copies of the files from this same location. When you click on **OK**, the **Setup** program copies the files and returns to the Options screen.

Note: If you are reinstalling the Remote API Client software in silent mode (as described in "Installing Remote API Client Software from the Command Line" on page 58), you may need to restart the computer to complete the installation. This is because some of the program files may be in use during the installation process (for example if the Remote API Client is running), and so cannot be replaced by the new files. In this case, the new files are copied to a temporary directory, and will be moved into place automatically when the computer is next restarted.

To check whether you need to restart the computer, use a text editor such as **Notepad** to view the contents of the installation log file when the installation process has completed. The installation log file is called **setup.log** and is created in the directory from which you run the Setup program, unless you use the **-f2** command-line option to specify a different path and filename.

At the end of the file, under the heading `Response Result`, the text `Result Code` should be followed by one of the two values `0` (zero) or `-12`. If the value is `0`, there is no need to restart the computer; if the value is `-12`, restart the computer before attempting to use the Windows Client.

Uninstalling the Remote API Client Software

You can uninstall the Remote API Client software at any time, by using the **Add/Remove Programs** option from the Windows Control Panel. Once the uninstall process is confirmed, Windows does the following:

- Deletes all installed files.
- If any **.DLL** files were saved to a subdirectory during the initial installation, restores files to their original location.
- Deletes the subdirectory in which the saved **.DLL** files were stored, as long as the subdirectory is empty.
- Removes the Program folder and created directory if they are empty.
- If the uninstallation is successful, deletes the **setup.log** file, which contains all file transfers and deletions.
- Displays a message saying either that the uninstallation was successful, or that the user should check the **setup.log** file because some part of the installation failed.

The **Exit** button returns you to Windows.

Uninstalling the Remote API Client Software from the Command Line

Instead of using the **Add/Remove Programs** option from the Windows Control Panel, you can uninstall the Remote API Client software from the command line. Use the following command:

installdir\sxcluninst -y

- *installdir* represents the directory in which you installed the client software, which is **C:\IBMCS\w32cli** (32–bit client) or **C:\IBMCS\w64cli** (64–bit client) unless you specified a different location during the client installation.
- The **-y** option is used for confirmation, to prevent accidental use of this command.

The uninstall program completes without prompts and does not require any further input.

Help

You can access Help at any time by pressing the **F1** key. The Configuration and Advanced Options windows each have **Help** buttons as well.

Chapter 8. Configuring and Using CS Linux

The easiest way to define and modify the CS Linux configuration is to use the Motif administration program (**xsnaadmin**). This program provides a graphical user interface from which you can view and manage SNA resources on the local node. You can also use other administration tools such as command-line administration, but the Motif program is recommended.

The Motif administration program includes help screens that provide overview information for SNA and CS Linux, reference information for CS Linux dialogs, and guidance for performing specific tasks. For each task (such as configuring the node) or type of communications (such as TN3270 or APPC), the program guides you in setting up the configuration of the required resources.

The Motif administration program enables you to set up all required parameters for standard CS Linux configurations. For advanced parameters, the Motif administration program supplies default values. You need to supply only the essential configuration information, which enables you to set up SNA communications quickly and easily.

You can also use the Motif administration program to manage the running CS Linux system. The administration program enables you to make and apply changes to the configuration while CS Linux is active, and provides easy access to status information for node resources.

The Motif administration program automatically displays status information for CS Linux resources. Most of this information is shown on the Node window (see "Managing CS Linux with the Motif Administration Program" on page 68). In addition, you can control certain resources—such as nodes and link stations—using the **Start** and **Stop** buttons on the Node window. Other resources are always started and stopped automatically, so there is no need to control them manually.

Note:

1. You must be a member of the login group `sna` to define or modify resources for CS Linux.
2. To use the Motif administration program, you must have an X-terminal.
3. For more information about the Motif administration program's user interface, including the buttons and icons that appear in its windows, refer to the program's help screens or to *IBM Communications Server for Data Center Deployment on Linux Administration Guide*.
4. The windows and dialogs in the Motif administration program may differ from those shown in this guide, depending on the choices you make on a particular dialog.

For information about other CS Linux administration tools, including command-line administration and NOF application programs, refer to *IBM Communications Server for Data Center Deployment on Linux Administration Guide*, *IBM Communications Server for Data Center Deployment on Linux Administration Command Reference*, or *IBM Communications Server for Data Center Deployment on Linux NOF Programmer's Guide*.

Planning for CS Linux Configuration

Before you make any configuration changes it is very important to plan thoroughly. Changes that you make can cause disruption, not only to the users of your local node but possibly to users all around the network.

You may find it useful to draw a diagram of any changes that you are making to the topology of the network. If you are adding or removing connections to other nodes, draw a picture showing your node and the other nodes. You can use the Motif administration program to gather configuration information about all of the existing connections and add that information to your diagram.

When you add new resources to your diagram, it is easy to see whether they duplicate existing ones, or whether any names clash. Similarly, your diagram can help you decide which resources you need to remove and help you avoid deleting essential ones.

If you are configuring a Client/Server CS Linux system with more than one node, ensure that you include all the CS Linux nodes and their connectivity resources in your diagram. You can then configure each node in turn as described in this chapter, in the same way as you would configure a standalone node.

Once you determine the changes you need to make, collect the configuration information that you need. To guide you in collecting configuration information for specific CS Linux functions, you can use the task sheets provided in the online help for the Motif administration program, or the planning worksheets provided in *IBM Communications Server for Data Center Deployment on Linux Administration Guide*.

This chapter provides instructions for configuring the most frequently used functions available in CS Linux. For each configuration task, this guide also notes the information you need to gather before configuring the resource.

Note: This guide does not provide detailed descriptions of the configuration information you need to enter on CS Linux dialogs. For more information about the fields on a particular dialog, consult the online help for that dialog in the Motif administration program.

Planning Worksheets

Before you begin to configure resources for CS Linux, gather all of the configuration data for the new resources. To record all of the information for a particular function or application that you need to support, use the planning worksheets provided in *IBM Communications Server for Data Center Deployment on Linux Administration Guide*.

You will probably need to gather configuration information from several sources, such as network administrators, host administrators, application programmers, and end users.

If you are trying to connect to another node, the administrator at that node is a key contact. The administrator for a node can tell you names, addresses and characteristics of all the resources on that node. Often, you will need to ensure that matching configuration parameters are entered at the local node and the remote node.

Task Sheets

The online help screens in the Motif administration program contain task sheets that provide guidance for specific configuration tasks. The task sheets contain pointers to all of the help screens for the dialogs that you will use to enter the configuration information. You can use these to browse the help and see exactly what data you must collect.

The task sheets also refer to more detailed help for each of the individual windows and dialogs that you must use to enter configuration information. Those help screens explain each field that you must fill in or select.

Setting environment variables for use by CS Linux

CS Linux uses a number of environment variables to control its operation, which must be set before you enable the CS Linux software on the server. The simplest way to manage these is to set them in a text file that CS Linux reads on startup.

In the CS Linux configuration directory, **/etc/opt/ibm/sna**, create a text file named **environment**, and set each environment variable in a separate line. For example:

```
export  LANG=en_US
export  PATH="$PATH:/opt/ibm/sna/bin"
export  LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
export  LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

If you are running 64-bit applications, change the environment variables as below:

```
export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64

export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64
```

Using the Motif Administration Program

Before you use the Motif administration program, you may want to add path information to your **.login** or **.profile** file to enable the system to find executable programs (see "Specifying the Path to CS Linux Programs"). In addition, you must enable the CS Linux software before you can use the administration program (see "Enabling CS Linux" on page 68).

For information about invoking the Motif administration program and an overview of using the program, see "Managing CS Linux with the Motif Administration Program" on page 68.

Specifying the Path to CS Linux Programs

To run CS Linux programs, you must specify the path to the directory that contains the CS Linux executable programs. You can specify the path either by adding the directory to your `PATH` environment variable before you run the programs for the first time, or by including the directory name each time you run the programs.

The Motif administration program is stored in the directory **/opt/ibm/sna/bin/X11**, and the other programs are stored in the directory **/opt/ibm/sna/bin**. If you add these directories to the definition of the `PATH` environment variable in your **.login** or **.profile** file, CS Linux locates the programs automatically. Alternatively, you can specify the directory name when you run the program, as in the following examples:

```
/opt/ibm/sna/bin/sna start
```

```
/opt/ibm/sna/bin/X11/xsnaadmin
```

The sample command lines shown in this manual assume that you have added the directories to your `PATH` environment variable, and do not include the directory names.

Enabling CS Linux

CS Linux must be enabled on the local system before you can configure or manage the local node. As with any X/Motif application, you may also need to set up the `DISPLAY` environment variable to indicate a suitable X server.

To enable CS Linux, enter the following command at the Linux command prompt:

```
sna start
```

Note: When you use the **sna start** command, the CS Linux software uses the directory from which you issued the command as its current working directory, and maintains one or more open file descriptors in that directory. This means that you will not be able to unmount the file system containing that directory while the CS Linux software is running. To avoid problems, you should start the CS Linux software from a directory on a filesystem that does not need to be unmounted; for example, you could use `cd /` to change to the root directory before using the **sna start** command.

When you install CS Linux, the installation utility automatically updates the startup file **/etc/rc.d/init.d/snastart** to include the **sna start** command. This ensures that CS Linux is started automatically at system startup. If you do not want CS Linux to be started automatically, you can remove or comment out this line, and then follow the instructions in this section to enable the CS Linux software manually.

CS Linux writes messages to standard error (normally your terminal's screen) to indicate that it is initializing, and to indicate whether initialization completes successfully.

Managing CS Linux with the Motif Administration Program

To use the Motif administration program for CS Linux, first make sure that CS Linux is initialized as described in “Enabling CS Linux.” (You may also need to set up the `DISPLAY` environment variable to indicate a suitable X server.)

To start the Motif administration program in the background, issue the following command:

```
xsnaadmin &
```

CS Linux displays the Domain window. This window shows all defined nodes, and enables you to start and stop nodes. Double-clicking on any node brings up the Node window for that node, as shown in Figure 5 on page 70.

The Node window shows information about the node and its resources. If you have not yet configured the node, the administration program prompts you to configure it as described in “Configuring the Node” on page 74.

Note: This guide uses the term window to describe Motif windows that display information about CS Linux resources. A window can contain one or more sections, or panes. A dialog is a Motif window on which you can enter information.

The Node window shows most of the information you need, and gives easy access to everything else. It shows all the key resources on the local node.

If you are configuring a Client/Server CS Linux system with more than one node, follow the instructions in this chapter to configure each node in turn (returning to the Domain window to select the next node).

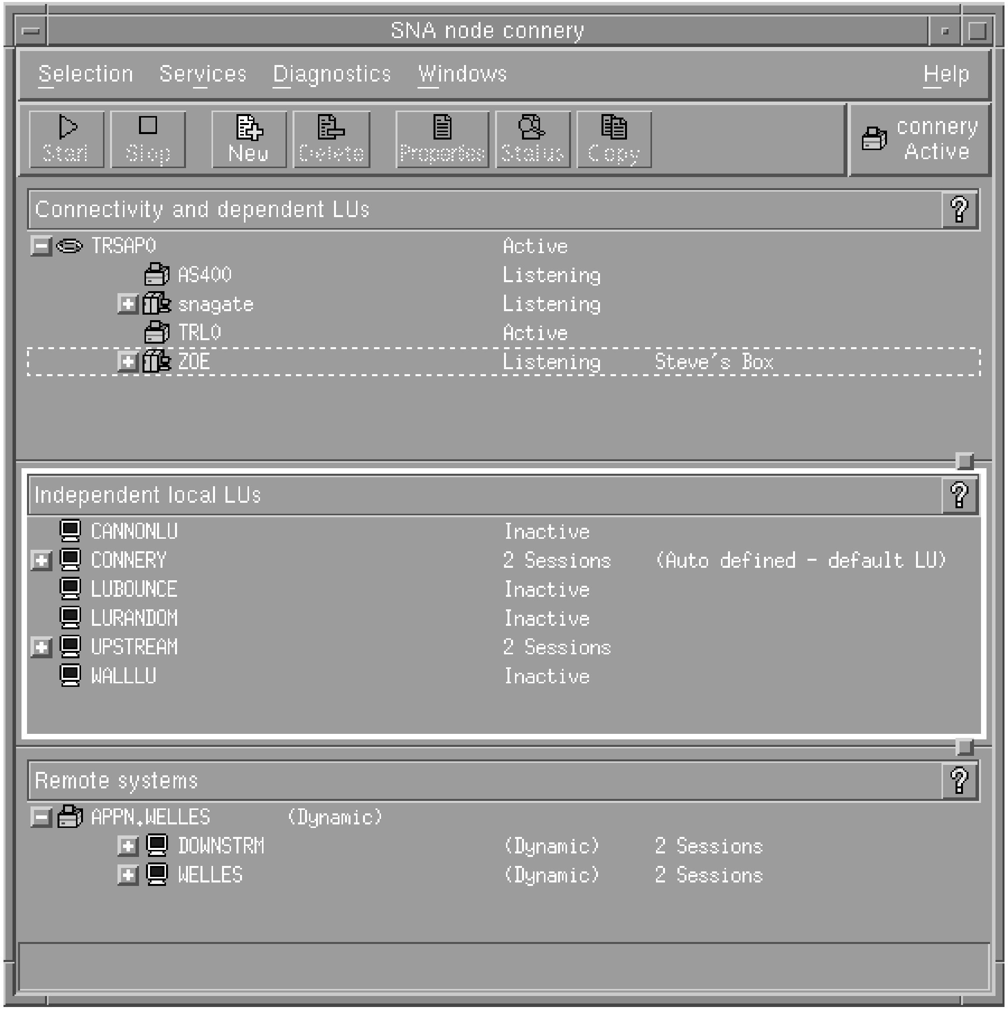
Other windows can be reached from the **Windows** menu in the Node window. These windows include the following:

- LU Pools window
- CPI-C Destination Names window

The **Services** menu in the Node window provides a quick way to add resources and provides help for configuration and management tasks. The **Diagnostics** menu takes you to the Logging dialog and Tracing dialog.

Node Window

A sample Node window is shown in Figure 5 on page 70. The title bar shows the name of the Linux system.



```
Connectivity and dependent LUs
TRSAP0                              Active
    AS400                           Listening
    snagate                         Listening
    TRL0                            Active
    ZOE                             Listening     Steve's Box

Independent local LUs
CANNONLU                            Inactive
CONNERY                             2 Sessions    (Auto defined - default LU)
LUBOUNCE                            Inactive
LURANDOM                            Inactive
UPSTREAM                            2 Sessions
WALLLU                              Inactive

Remote systems
APPN.WELLES        (Dynamic)
    DOWNSTRM                        (Dynamic)     2 Sessions
    WELLES                          (Dynamic)     2 Sessions
```

Figure 5. Node Window

From the Node window, you can configure and manage all of the resources and components for the CS Linux node.

- Ports
- Link stations
- LUs of type 0-3 and dependent LUs of type 6.2
- DLUR internal PUs
- Independent local LUs
- Remote nodes
- Partner LUs

You can add, delete, modify and manage all of these resources from the Node window. The layout of the resources in the window shows the relationships among resources and enables you to control which resources are displayed.

Ports, local LUs, and remote nodes are always displayed. The Node window shows each link station below its parent port, and each dependent LU below its parent link station. It also shows partner LUs below local LUs and below remote nodes.

The Node window contains separate sections for the different types of resources for the node:

- The Node box in the top-right corner of the Node window indicates whether the node is `Active` or `Inactive`.
- The top pane of the Node window (the Connectivity pane) lists connectivity resources for the node, including ports, link stations or PUs on each port, and dependent LUs on a specific link station or PU. For each resource, this window shows current status information.
- The middle pane (the Independent Local LUs pane) shows independent LUs defined on the local node. This window also displays information about sessions using a particular LU, and any records that define a partner LU's location by the link station that is used to access it.
- The lower pane (the Remote Systems pane) shows information about remote nodes and partner LUs. It also shows session information for each remote node or partner LU.

You can select any of these panes by clicking on the pane. You can also select specific resources within a pane by clicking on the line for the resource. To view or modify the configuration for an item, you can double-click on the item. (You can use the buttons and menus on this window to access configuration information for specific resources.)

For each item listed, resources that belong to that item are nested within the information for that item. For example, link stations are grouped under the port to which they belong. You can click on the **Expand** button (+) next to an item to show the resources for that item if they are not currently displayed, or click on the **Contract** button (−) to hide the resources for an item.

You can perform the following administration tasks from the Node window:

Start or stop a resource
Select the resource and click on the **Start** or **Stop** button. (Alternatively, you can select **Start item** or **Stop item** from the **Selection** menu.)

Add a resource for an item
Select the item and click on the **New** button (or select **New** from the **Selection** menu). For example, to add a link station for a port, select the port and click on the **New** button.

Delete a resource
Select the resource and click on the **Delete** button (or select **Delete** from the **Selection** menu).

View or modify the configuration for any resource
Select the resource and click on the **Properties** button (or select **Properties** from the **Selection** menu).

Get status information for any resource
Select the resource and click on the **Status** button (or select **Status** from the **Selection** menu).

Copy the configuration for any resource
Select the resource and click on the **Copy** button (or select **Copy** from the **Selection** menu).

In addition, you can choose specific configuration tasks for the node from the **Services** menu, control logging (for the domain) and tracing (for the node) from the **Diagnostics** menu, and view or modify domain resources by selecting one of the items on the **Windows** menu.

Resource Items

The layout of the resources in a window shows the relationships among them.

If an item has one or more child items associated with it, an **Expand** symbol (+) or **Contract** symbol (–) appears next to it:

- An **Expand** symbol indicates that the associated child items are hidden. You can click on the **Expand** symbol, or press the **+** key on the numeric keypad, to show them.
- A **Contract** symbol indicates that the child items are shown. You can click on the **Contract** symbol, or press the **–** key on the numeric keypad, to hide them.
- If an item has neither symbol next to it, the item has no associated child resources.

For example, a link station is associated with a particular port. In the Connectivity pane of the Node window, the link station is displayed below its parent port, along with all other link stations associated with that port. The port is always displayed, but you can choose whether the list of associated link stations is shown or hidden. Similarly, link stations with a list of associated LUs can be expanded to show the LUs, or contracted to hide them.

A parent resource must always be configured before its child resources, and deleting the parent resource causes all its child resources to be deleted too.

Tool Bar Buttons

Resource windows include tool bar buttons to make it easy to perform common functions. A tool bar for CS Linux is shown in Figure 6.

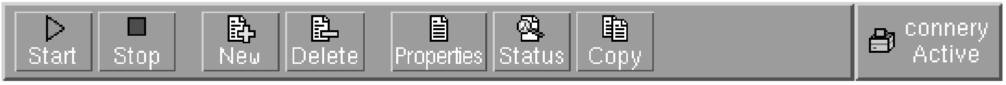


Figure 6. CS Linux Tool Bar

Not all buttons appear in the tool bars of each resource window. If a button's operation is not valid for the currently selected item (or an operation requires an item to be selected, but none is), the outline of the button is displayed in gray, and the function cannot be selected (clicking on the button has no effect). The following buttons can appear on resource windows:

Start Starts the selected item.

Stop Stops the selected item.

New Adds a new resource item.

Delete Deletes the selected resources.

Properties
Opens the dialog for the selected item to view or modify the item's configuration.

Status Displays the current status of the selected item.

Copy Copies the selected item. Clicking on this button opens a dialog whose fields duplicate the configuration of the selected item. Complete the dialog's fields (filling in the new item's name) to add the new resource.

Many resources, such as ports and link stations, cannot be modified while they are active. You can, however, view an active resource's parameters by selecting the resource and clicking on the **Properties** button to open its dialog. Click on the **Close** button when you are finished.

Configuring Client/Server Functions

This section is relevant only if you installed CS Linux to run in a client/server environment (with multiple CS Linux nodes in the same network).

In a client/server environment, a server can be marked as a configuration server; CS Linux maintains a list of these configuration servers. The first server listed is the master server, and any other servers listed are backup servers. The servers are listed in order, so that the second server listed (the first backup server) takes over if the master server is unavailable, the third server listed (the second backup server) takes over if neither the master nor the first backup server is available, and so on.

When any of the nodes in the domain are active, the first available configuration server in the domain (the first server that can be contacted and has CS Linux software running) becomes the master server. If the current master becomes unavailable (because it cannot be contacted, perhaps due to a network failure, or because the SNA software running on it is stopped), the next available configuration server in the list becomes the new master.

CS Linux can run without a master. This happens if none of the servers in the configuration server list can be contacted. If this happens, you can view and configure node resources only on the servers that can be contacted.

Note: You cannot directly indicate which node acts as the master server; the master server is selected based on the order in which nodes are added to the configuration server list. If you wish to move a server to the top of the list, remove all other nodes from the list and then add them again.

In the Motif administration program Domain window, you can add a configuration server by selecting **Make configuration server** from the **Selection** menu. The server is added to the end of the list; it becomes the master server only if all other configuration servers are unavailable. To remove a server, select **Remove configuration server** from the **Selection** menu.

Note: You cannot delete a server if it is the only server listed on which the CS Linux software is running, because in this case there is no other server that can take over as the master server. At least one enabled master server is required in a client/server configuration.

For more information about configuring and managing a Client/Server CS Linux system, refer to *IBM Communications Server for Data Center Deployment on Linux Administration Guide*. This manual also provides information about advanced Client/Server configuration, including how to move clients and servers into different CS Linux domains and how to configure the details of client operation.

Configuring the Node

The first step in configuring CS Linux on a system is to configure the local node. Node configuration provides the basic information that the node needs in order to communicate in an SNA network. You must configure the node before you can define connectivity or other resources for the node.

If the node has already been configured, you can use the procedures described in this section to modify the node configuration; but you must stop the node before making configuration changes.

You can configure the CS Linux node as either an APPN network node or an APPN end node (if CS Linux uses SNA only for communication with the host, you probably want to configure the CS Linux node as an end node or branch network node).

When the local node is part of an APPN network, configure it as an APPN network node if the node is to provide APPN routing services for other nodes. If other nodes provide routing services, configure the local node as an APPN end node.

Before you begin the node configuration, gather the following information:

- Type of APPN support (network node, branch network node, end node).
- Control point name (and alias, if different). Consult with your network planner to determine this name.
- Default Node ID. (You can override this default when configuring an individual communications link.)

To configure the node, perform the following steps from the Node window:

1. Select **Configure node parameters** from the **Services** menu, or double-click on the Node box in the top-right corner of the Node window. CS Linux displays the Node Parameters dialog.
2. Specify the level of APPN support, the control point name, and (if necessary) the default node ID.
3. Click on the **OK** button to define the node. When you define the node, CS Linux automatically defines a default LU with the same name as the control point.

   To exit without saving the values you have entered, click on the **Cancel** button.

Configuring Connectivity

For a CS Linux node to communicate with other nodes, you must configure connectivity with at least one adjacent node. A connecting link can be configured to carry dependent traffic, independent traffic, or both.

You can have adapter cards for one or more link protocols installed in your computer. Much of the information you need to enter to configure connectivity depends on the link protocol you are using. For a list of the link protocols supported by CS Linux, see “Installation Requirements” on page 19.

To configure a link, you need to define a port and (in most cases) a link station. When using the Motif administration program, a DLC (data link control) is automatically configured as part of the configuration for the port. In addition, you have the option of defining the port as part of a connection network.

The links that you need to configure depend on what you are trying to achieve, and on whether your network is an APPN network. The information required depends on the link protocol, and on whether the link is for dependent traffic, independent traffic, or both.

As examples, this section explains how to configure the following types of links:

- Link supporting dependent traffic with a host system using an SDLC line.
- Link supporting both dependent and independent traffic into an APPN network using the Ethernet link protocol. This example also defines a connection network on the Ethernet port.
- Enterprise Extender link into an APPN network (note that Enterprise Extender links support only independent traffic).

For other link protocols, refer to *IBM Communications Server for Data Center Deployment on Linux Administration Guide* or the online help for the Motif administration program.

Configuring an Ethernet Link to Support Dependent and Independent Traffic

This example shows how to configure an Ethernet link supporting both dependent and independent traffic into an APPN network. In addition, it defines a connection network on the Ethernet port.

For an Ethernet port, you need the following information:

- SNA port name (you can generally use the default). If you have multiple Ethernet network adapter cards, you also need to supply the Ethernet card number. You also need to specify the local SAP (service access point) number (normally `04` for Intel and OSA2 adapters). For an OSA-Express adapter, the local SAP number must match that defined in the OSA/SF for the I/O device addresses that correspond to the ethX interface on this Linux image.
- Whether the port should activate automatically when the node is started.
- Connection network name (must be the same on all ports in the same connection network).

For an Ethernet link station, you need the following additional information:

- Activation method (by administrator, on node startup, or on demand).
- Type of traffic supported (for this example, both dependent and independent).
- Remote node control point name (only needed for a LEN node).
- Remote node type (network node, end node, or discover).
- Remote node role (for this example, downstream SNA gateway or passthrough DLUR).
- To configure a selective link station, you need the MAC (medium access control) address and the SAP number (normally `04`) for the remote station. If you do not supply address information and you specify `By administrator` for the *Activation* field, the link station is a nonselective listening link station.

To configure the Ethernet link, perform the following steps from the Node window:

1. Configure the port:
   a. Select the Connectivity pane of the window.

   b. Select **New port** from the **Connectivity** submenu on the **Services** menu (or click on the **New** button in the button bar).
   c. On the resulting dialog, select the protocol type from the option menu, then choose to define a port.

      When you click on the **OK** button, CS Linux displays the Ethernet SAP dialog.
   d. Enter appropriate values in the fields on the dialog.
   e. Click on the **OK** button to define the port.

      The port appears in the Connectivity pane of the Node window.
2. Define a link station on the port:
   a. Make sure you have selected the port to which the link station is being added in the Connectivity pane of the Node window.
   b. Select **New link station** from the **Connectivity** submenu on the **Services** menu (or click on the **New** button in the button bar).
   c. Click on the **OK** button.

      CS Linux displays the Ethernet Link Station dialog.
   d. Enter appropriate values in the fields on the dialog.
   e. Click on the **OK** button to define the link station.

      The link station appears beneath the port to which it belongs in the Connectivity pane of the Node window.

Configuring an Enterprise Extender Link

This example shows how to configure an Enterprise Extender link into an APPN network. Note that Enterprise Extender links support only Independent LU traffic.

For an Enterprise Extender port, you need the following information:
- SNA port name (you can generally use the default). If you have multiple network adapter cards running IP, you also need to supply the IP interface name you want to use (such as `eth0`).
- Whether the port should activate automatically when the node is started.

For an Enterprise Extender link station, you need the following additional information:
- Activation method (by administrator, on node startup, or on demand).
- Remote node type (network node, end node, or discover).
- To configure a selective link station, you need the IP hostname or IP address for the remote station. If you do not supply this information and you specify `By administrator` for the *Activation* field, the link station is a nonselective listening link station.

To configure the Enterprise Extender link, perform the following steps from the Node window:
1. Configure the port:
   a. Select the Connectivity pane of the window.
   b. Select **New port** from the **Connectivity** submenu on the **Services** menu (or click on the **New** button in the button bar).
   c. On the resulting dialog, select the protocol type from the option menu, then choose to define a port.

      When you click on the **OK** button, CS Linux displays the IP Port dialog.

- d. Enter appropriate values in the fields on the dialog.
- e. Click on the **OK** button to define the port.

  The port appears in the Connectivity pane of the Node window.

2. Define a link station on the port:
   - a. Make sure you have selected the port to which the link station is being added in the Connectivity pane of the Node window.
   - b. Select **New link station** from the **Connectivity** submenu on the **Services** menu (or click on the **New** button in the button bar).
   - c. Click on the **OK** button.

     CS Linux displays the IP Link Station dialog.
   - d. Enter appropriate values in the fields on the dialog.
   - e. Click on the **OK** button to define the link station.

     The link station appears beneath the port to which it belongs in the Connectivity pane of the Node window.

Configuring Type 0–3 LUs

To support user applications that use type 0–3 LUs, you must configure dependent LUs. Before doing so, you must perform the following configuration:

- Configure the node as described in "Configuring the Node" on page 74.
- Configure a link to support dependent LU traffic as described in "Configuring Connectivity" on page 74.

  You do not need to configure a direct link to the host if you have an upstream link to another node using SNA gateway, or if you are using DLUR. For more information, see "Configuring SNA Gateway" on page 86 and "Configuring DLUR" on page 88.

You must configure dependent LUs of types 0–3 to support communication with a host system. You can use the information in this section to define an LU to support LUA, DLUR, or PU Concentration. You can also define a range of LUs, to configure multiple LUs of the same type in a single operation.

In addition, you can define a pool of LUs to be used as required, either by assigning an LU to a pool when you define the LU or by assigning previously defined LUs to a pool.

Defining Type 0–3 LUs

Before configuring the 3270 LU, gather the following information:

- LU name. (This is a local identifier, and does not have to match the host configuration.)
- LU number (or numbers for a range of LUs).
- LU type (3270 display model or 3270 printer).
- Pool name (if you are adding the LU to a pool).

To configure an LU of types 0–3 for a previously defined link station, perform the following steps from the Node window:

1. Select the link station to the host in the Connectivity pane of the window.
2. Click on the **New** button.
3. Select the LU type (**New 3270 display LU** or **New 3270 printer LU**) on the resulting dialog.

When you select this item and click on **OK**, CS Linux displays the LU Type 0–3 dialog.

4. Enter appropriate values in the fields on the dialog.
5. Click on **OK** to define the LU.

   The LU appears in the Connectivity pane of the Node window, below the link station to the host.

Defining an LU Pool

For LU type 0–3, you can define LU pools to simplify user configuration and provide greater flexibility in establishing host sessions. For example, you can define several LUs in a single LU pool, then configure multiple users using this LU pool. This makes configuring the users' sessions easier and enables any session to use any LU in the pool.

Note: You can assign a user's session either to a specific LU or to an LU pool.

- If you assign the user's session to a specific LU that is in a pool, the session uses this LU if it is available; otherwise it uses any free LU from the pool, as though you had assigned it to the LU pool instead of the specific LU.
- If you want the user to use only a specified LU, so that the user's session cannot be established if the LU is already in use, ensure that the LU is not in a pool.

You can view the LU pools for the local CS Linux node using the LU Pools window. This window lists the LU pools configured on the local system, and enables you to select LUs to add to an LU pool.

You can add the following LU types to a pool (do not mix LUs of different types in the same pool):

- 3270 display LU
- Unrestricted LU

Before you can add LUs to a pool, the LUs must be defined on the local node.

To configure an LU pool, perform the following steps from the Node window:

1. Select **LU Pools** from the **Windows** menu.

   CS Linux displays the LU Pools window.
2. Click on the **New** button.

   CS Linux displays the LU Pool Configuration dialog.

   The box on the right lists LUs that are not yet allocated to any pool. Any of these LUs can be included in the new pool.
3. Select the LU or LUs you wish to add to the pool, and click on the **New** button to move the selected LUs to the box on the left.

   To remove an LU from the box on the left, select it and click on the **Remove** button.
4. Click on **OK** to define the LU pool.

   All of the LUs in the box on the left are added to the LU pool.

   The pool appears in the LU Pools window.

Configuring APPC Communication

APPC applications and CPI-C applications require that you configure APPC first. An APPC application uses the node's LU type 6.2 resources to communicate with another APPC or CPI-C application on a host or peer computer, using a specified mode.

Before you can configure APPC communication, you must perform the following configuration:

1. Configure the node as described in “Configuring the Node” on page 74.
2. Configure connectivity as described in “Configuring Connectivity” on page 74.

The remaining configuration steps depend on whether the configuration supports dependent traffic, independent traffic, or both:

Independent APPC

Independent APPC uses independent LUs. Each LU-LU session involves a local LU and a partner LU.

For the local LU, you can use the predefined default LU associated with the node control point, or you can configure new local LUs.

The partner LU need not be configured at all if the CS Linux node is an end node or network node in an APPN network, because APPN can locate partner LUs dynamically. However, you do have to configure the partner LU if your network is not an APPN network or if the node is a LEN node. In this case, you must configure the remote node where the partner LU resides, then define the partner LU on the remote node.

Dependent APPC

If the remote node is a host that does not support independent LU 6.2, configure for dependent traffic. For dependent APPC, you must configure a local LU.

If the applications use CPI-C, you may need to do additional CPI-C configuration after configuring APPC (see “Configuring for CPI Communications” on page 84). A CPI-C application uses the node's LU type 6.2 and mode resources to communicate with another APPC or CPI-C application on a host or peer computer. You define the same resources for a CPI-C application as for an APPC application. In addition, if the TP on the CS Linux computer is the invoking TP, also known as the source TP (the TP that starts the conversation), you may need to define one or more side information entries for it, as described in “Configuring for CPI Communications” on page 84. Each of these entries provides information on a partner TP, the LU and mode resources used to access it, and any security information required.

This section explains how to configure a simple APPN network (using independent LU 6.2) that consists of a network node, an end node, and a LEN node, as described in “Configuring a Simple APPN Network” on page 80. (This scenario also shows how you can get status information for CP-CP sessions between two nodes.)

This section also explains how to configure dependent APPC communication, as described in “Configuring Dependent APPC” on page 84.

Both of these scenarios assume that APPC sessions use a standard mode and class-of-service (COS).

For information about configuring additional APPC information, such as modes, security, and invokable (target) TPs, refer to *IBM Communications Server for Data Center Deployment on Linux Administration Guide*.

Configuring a Simple APPN Network

The simplest APPN network you can configure includes only two nodes: an APPN network node and an APPN end node. The network node handles session routing for the end node.

Configuring a Network Node

This scenario assumes that you are using the control point LU and a standard mode, and that you are using a LAN link type (Token Ring, Ethernet). In this case, you can configure the network node simply by performing the following configuration tasks:

1. Configure the node as described in “Configuring the Node” on page 74. For the *APPN support* field, select the value `Network node`. Make a note of the control point name.
2. Configure connectivity as described in “Configuring Connectivity” on page 74. Configure the link to support independent traffic.

To contact this network node from an adjacent end node, you'll need to know the MAC address and SAP number of the port on the network node. You can use the following procedure to get the MAC address on a CS Linux node:

1. Select the port on the Node window.
2. Click on the **Start** button to start the port.
3. Click on the **Status** button to get status information for the port. The Port Status dialog shows the MAC address and SAP number.
4. Make a note of the MAC address and SAP number so you can enter those values on the link station configuration dialog for the end node.

Configuring an End Node

This scenario assumes that you are using the control point LU and a standard mode, and that you are using a LAN link type (Token Ring, Ethernet). In this case, you can configure the network node simply by performing the following configuration tasks:

1. Configure the node as described in “Configuring the Node” on page 74. For the *APPN support* field, select the value `End node`.
2. Configure connectivity as described in “Configuring Connectivity” on page 74. Configure the link to support independent traffic, and supply the following information for the link station:
   - Enter the name of the network node (see “Configuring a Network Node”) as the value for the *Remote node* field.
   - Enter the MAC address and SAP number for the port on the network node in the Contact Information pane on the link station configuration dialog.

   In an APPN network, a single link station to an adjacent network node can be used to communicate with any remote node in the network, so you do not need to configure a separate link station to each remote node.

Verifying Connectivity between Two Nodes

This scenario assumes that you have configured a network node as described in “Configuring a Network Node,” and an end node as described in “Configuring an End Node.” You can perform the following procedure from the end node:

1. On the Node window, select the link station that connects to the adjacent network node.
2. Click on the **Start** button to start the link station.

   When the link station is started, the CP-CP sessions between the two nodes are established automatically. Those sessions are displayed in the Independent Local LUs pane of the Node window.
3. To get status information for a session, select the session on the Node window and click on the **Status** button.

Configuring an Independent APPC LU

In many cases, applications can use the local node's control point LU, which is automatically defined when you configure the node. This is the default LU—if your application does not specify a particular LU, it can use this one. If the application uses the default LU, you do not need to define a local LU. Check the documentation for your APPC application, or contact the application programmer.

To configure an independent LU 6.2, you need the following information:

- Local LU name.
- Local LU alias (if an alias is used in a TP that this LU supports).

To configure an independent local LU, perform the following steps from the Node window:

1. Select the Independent Local LUs pane of the window.
2. Select **New independent local LU** from the **APPC** submenu on the **Services** menu (or click on the **New** button).

   CS Linux displays the Local LU dialog.
3. Enter appropriate values in the fields on the dialog.
4. Click on the **OK** button to define the local LU. The independent LU appears in the Independent Local LUs pane of the Node window.

Configuring Partner LUs for a LEN Node

You must define a remote node (and the partner LUs on the node) in the following situations:

- If the local node is a LEN node, you must define all of the remote nodes and any partner LUs on the remote node with which it communicates using APPC. A LEN node is not able to dynamically locate partner LUs; the remote node definition enables it to do so.
- If the local node is not part of an APPN network (for example, if you have two end nodes directly connected, with no network node server), LUs cannot be located dynamically. In this case, you must configure each partner LU.
- If the remote node is a LEN node and the local node is a network node that acts as the LEN node's network node server, you must define the LEN node (and its partner LUs) as a remote node on the network node server. This definition enables nodes in the rest of the APPN network to locate LUs on the LEN node.
- If the remote node is in a different APPN network, you must define the remote node because it cannot be dynamically located.

Do not define partner LUs if both the local and remote nodes are part of the same APPN network.

When you add a remote node definition, a partner LU with the same name as the remote node is automatically added; this is the control point LU for the remote node. If your application uses this partner LU, you do not need to add another partner LU, although you may want to add an LU alias for the partner LU. To add an alias, double click on the partner LU and enter the alias in the Partner LU Configuration dialog.

If your application uses an LU alias to refer to its partner LU, you should add a partner LU alias definition.

If either the local node or the remote node is a LEN node, you must define the partner LU as a child of the remote node, because a LEN node cannot take part in dynamic location of LUs. If your application uses the control point LU of the remote node as its partner LU, the control point LU was defined automatically when you defined the remote node.

You can use the Motif administration program to add a partner LU alias (see "Defining a Partner LU Alias"), add a definition of a partner LU on a specific remote node (see "Defining a Partner LU on a Remote Node"), or define multiple partner LUs using wildcards (see "Defining Multiple Partner LUs Using Wildcards" on page 83).

Defining a Remote Node: Before configuring a remote node, you need the following information:

- Fully qualified SNA network name of the node.

To configure a remote node, perform the following steps from the Node window:

1. Select the Remote Systems pane of the window.
2. Select **New remote node** from the **APPC** submenu on the **Services** menu (or click on the **New** button in the button bar, then select **Define remote node**).

   CS Linux displays the Remote Node Configuration dialog.
3. Enter appropriate values in the fields on the dialog.
4. Click on the **OK** button to define the remote node. The remote node appears in the Remote Systems pane of the Node window.

   When you define a remote system, CS Linux automatically defines the control point LU on the remote node as a partner LU on the local node.

Defining a Partner LU Alias: To define a partner LU alias, you need the following information:

- Fully qualified partner LU name (SNA network name and LU name)
- Partner LU alias used by a local TP

To add a partner LU alias, perform the following steps from the Node window:

1. Select the Remote Systems pane of the window.
2. Select **APPC**, **New partner LUs**, and **Partner LU alias** from the **Services** menu (or click on the **New** button in the button bar, and select **Define partner LU alias**).

   CS Linux displays the Partner LU Alias Configuration dialog.
3. Enter the partner LU name and alias on the dialog.
4. Click on the **OK** button to define the partner LU alias. The partner LU alias appears in the Remote Systems pane of the Node window (as part of the Network definition).

Defining a Partner LU on a Remote Node: To define a partner LU on a specific remote node, you need the following information:

- Fully qualified partner LU name.
- Partner LU alias (if an alias is used by a local TP).
- Fully qualified name of the node that contains directory information for the partner LU.

To add a partner LU definition for a specific remote node, perform the following steps from the Node window:

1. Select the remote node.
2. Select **APPC**, **New partner LUs**, and **Partner LU on remote node** from the **Services** menu (or click on the **New** button in the button bar, and select **Define partner LU on remote node**).

   CS Linux displays the Partner LU Configuration dialog.
3. Enter the appropriate values in the fields on the dialog.
4. Click on the **OK** button to define the partner LU. The partner LU alias appears in the Remote Systems pane of the Node window, under the remote system to which it belongs.

Defining Multiple Partner LUs Using Wildcards: You can use wildcards to configure the location for a set of partner LUs that are all located on the same remote node and whose names start with the same characters. Using wildcards means that you do not need to configure each partner LU individually.

When you define partner LUs using wildcards, you must supply the following information:

- Wildcard partner LU name. The wildcard partner LU name consists of two type A EBCDIC strings, each of 1–8 characters, that match the fully qualified LU names of multiple partner LUs.

  The first string can be a complete SNA network name that matches the network name for the partner LUs exactly, or a wildcard prefix that matches the beginning of the network name. If you enter a wildcard prefix for the network name, leave the second string blank.

  If you supply a complete SNA network name for the first string, you can also enter a value for the second string. (You cannot enter the second string unless you supplied a valid SNA network name for the first string.) The second string is treated as a wildcard prefix, which must match the start of the second part of the fully qualified partner LU names.
- Name of the node where the partner LUs are located.

To add multiple partner LUs, perform the following steps from the Node window:

1. Select the remote node for which you are defining the partner LUs.
2. Select **APPC**, **New partner LUs**, and **Wildcard partner LUs on remote node** from the **Services** menu (or click on the **New** button in the button bar, and select **Define wildcard partner LUs on remote node**).

   CS Linux displays the Wildcard Partner LU Configuration dialog.
3. Enter the appropriate information in the fields on the dialog.
4. Click on the **OK** button to define the partner LUs. The partner LUs appear in the Remote Systems pane of the Node window, under the remote node to which they belong.

Configuring Dependent APPC

To configure a dependent LU 6.2, you need the following information:

- Local LU name.
- Local LU alias (if an alias is used in a TP that this LU supports).
- Name of the link station that provides the connection to the host.
- LU number.
- Whether the LU should be assigned to the default pool for dependent LU 6.2.

If you are configuring dependent LUs of type 6.2 for use with APPC or CPI-C applications, you may wish to define them as members of the default pool. An application that does not specify a particular local LU is assigned an unused LU from the pool of LUs defined as default LUs.

To configure a dependent local LU, perform the following steps from the Node window:

1. Select a link station in the Connectivity pane of the window.
2. Select **New dependent local LU** from the **APPC** submenu on the **Services** menu (or click on the **New** button in the button bar, and select **New dependent local LU**).

   CS Linux displays the Local LU dialog.
3. Enter appropriate values in the fields on the dialog.
4. Click on the **OK** button to define the local LU. The dependent LU appears in the Connectivity pane, below the link station to which it belongs.

Configuring for CPI Communications

If you are supporting a CPI-C application that uses CPI-C symbolic destination names, you need to define the CPI-C side information. The side information associates the symbolic destination name with information about the partner TP, partner LU, mode, and security for the conversation.

To determine the symbolic destination name for CPI-C, consult the application developer (or for a third-party application, consult the product documentation).

Before configuring CPI-C side information, you need the following information:

- Symbolic destination name used by the TP
- Partner TP name
- Partner LU name or alias
- Mode name

To configure CPI-C side information, perform the following steps from the Node window:

1. Select **CPI-C** from the **APPC** sub-menu in the **Services** menu.

   CS Linux displays the CPI-C Destination Names window.
2. Click on the **New** button.

   CS Linux displays the CPI-C Destination Configuration dialog.
3. Enter appropriate values in the fields on the dialog.
4. Click on the **OK** button to define the CPI-C side information.

Configuring LUA

The LUA API can be used for applications that use LU types 0–3 to communicate with a host computer. (For detailed information about the LUA API, refer to *IBM Communications Server for Data Center Deployment on AIX or Linux LUA Programmer's Guide*.)

Before configuring LUA, perform the following configuration:

1. Configure the node as described in “Configuring the Node” on page 74.
2. Configure connectivity for dependent traffic as described in “Configuring Connectivity” on page 74. (If you are using upstream SNA gateway or DLUR, configure the link to the upstream node instead of a direct link to the host.)

To configure LUA, you need the following information:

- LU name or LU pool name.
- LU number for each LU. The LU number must match the LU number configured on the host.

To configure LUA, define the LU using the following procedure:

1. Select the link station to the host in the Connectivity pane of the Node window.
2. Click on the **New** button.
3. On the resulting dialog, select **New LU for LUA**.
4. Enter appropriate values in the fields on the dialog. Specify an LU type of `Unrestricted`.
5. Click on the **OK** button. The LU appears in the Connectivity pane of the Node window, beneath the link station to the host.
6. If you are going to use any LU pools, define them as described in “Defining an LU Pool.”

Defining an LU Pool

You can define LU pools to simplify user configuration and provide greater flexibility in establishing host sessions. For example, you can define several LUs in a single LU pool, then configure multiple LUA applications using this LU pool. This makes configuring the applications easier and enables any application to use any LU in the pool.

Note: You can assign a user's session either to a specific LU or to an LU pool.

- If you assign the user's session to a specific LU that is in a pool, the session uses this LU if it is available; otherwise it uses any free LU from the pool, as though you had assigned it to the LU pool instead of the specific LU.
- If you want the user to use only a specified LU, so that the user's session cannot be established if the LU is already in use, ensure that the LU is not in a pool.

You can view the LU pools for the local CS Linux node using the LU Pools window. This window lists the LU pools configured on the local system, and enables you to select LUs to add to an LU pool.

You can add the following LU types to a pool for use by 3270 (do not mix LUs of different types in the same pool):

- 3270 display LU

- Unrestricted LU

Before you can add LUs to a pool, the LUs must be defined on the local node.

To configure an LU pool, perform the following steps from the Node window:

1. Select **LU Pools** from the **Windows** menu.

   CS Linux displays the LU Pools window.
2. Click on the **New** button.

   CS Linux displays the LU Pool Configuration dialog.

   The box on the right lists LUs that are not yet allocated to any pool. Any of these LUs with type `Unrestricted LU` can be included in the new pool for LUA.
3. Select the LU or LUs you wish to add to the pool, and click on the **New** button to move the selected LUs to the box on the left.

   To remove an LU from the box on the left, select it and click on the **Remove** button.
4. Click on **OK** to define the LU pool.

   All of the LUs in the box on the left are added to the LU pool.

   The pool appears in the LU Pools window.

Configuring SNA Gateway

In addition to providing direct access to a host computer, CS Linux can provide SNA gateway facilities. This feature enables other computers to access a host computer through a CS Linux node, instead of requiring a separate connection to the host from each computer.

The SNA gateway feature is shown in Figure 7.

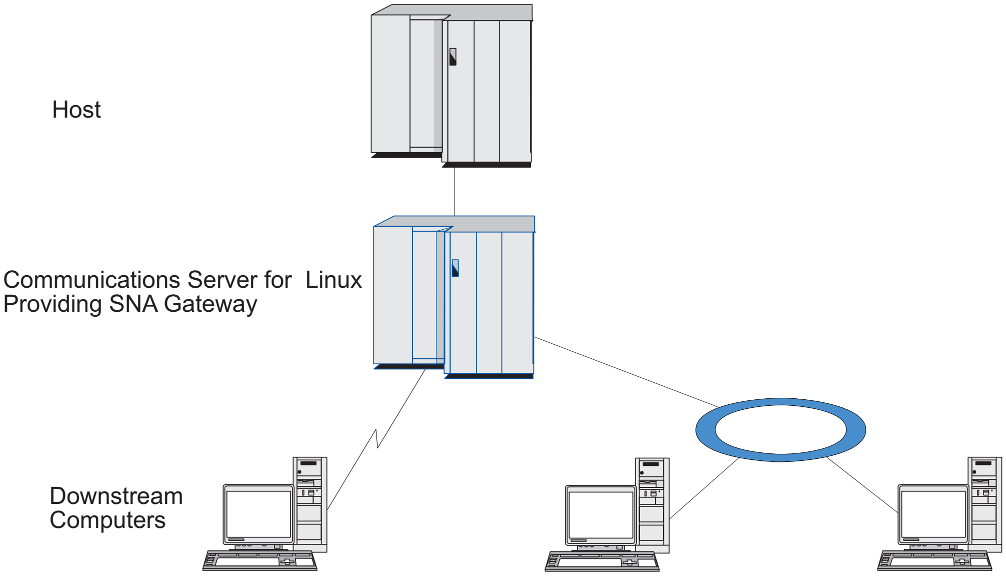


Figure 7. SNA Gateway

The downstream computer must contain an SNA PU type 2.0 or 2.1 to support dependent LUs. For example, the downstream computer could be another CS Linux computer or a PC running Communications Server for Windows.

When the local CS Linux node uses the SNA gateway feature, all the data transferred between the host and the downstream computer is routed through the local node. This enables a downstream computer to share a host connection with CS Linux or with other downstream computers, instead of requiring a direct link. For example, you could set up several downstream computers connected to CS Linux over a local token ring network, so that they could all access the same long-distance leased line from CS Linux to the host.

Using SNA gateway also simplifies the configuration at the host, because you do not need to define the downstream computers and the communication links to them. The host configuration needs to include only the CS Linux computer and its host communication link; the LUs at the downstream computers are configured as part of the resources of the CS Linux computer. The host computer is not aware that SNA gateway is being used.

Before configuring SNA gateway, you must perform the following configuration tasks:

- Define the local node as described in “Configuring the Node” on page 74.
- Configure a port and link station for dependent traffic between the local node and the host, as described in “Configuring Connectivity” on page 74. Also, configure ports and link stations for dependent traffic between the local node and the downstream nodes. If you need to support downstream LUs that are not defined in advance, you can define a template on the port to support implicit PUs and downstream LUs (see “Supporting Implicit Downstream LUs”).
- Define the LUs on the local node that are used for communication with the host (upstream LUs). Define the upstream LUs as LU type 0–3 with an LU type of `Unrestricted` (`unknown`). (The LUs on downstream nodes can be any LU type.)
- If you are going to use any LU pools, define them as described in “Defining an LU Pool” on page 78.

Supporting Implicit Downstream LUs

To support downstream LUs that are not predefined to CS Linux, you can define a template on the port for implicit downstream PUs and LUs (for basic port configuration, see “Configuring Connectivity” on page 74). These templates provide support for downstream LUs without requiring that you configure an LU on the local node to support every LU on a downstream node.

Before configuring a downstream LU for SNA gateway, you need the following information:

- Range of LU numbers to support downstream LUs.
- Host LU name.

To define a template for implicit downstream LUs, perform the following steps:

1. If you have already configured the port, double-click on the port definition in the Connectivity pane of the Node window. CS Linux displays the port configuration dialog.

   If you have not already configured the port, do so now:

   a. Select the Connectivity pane on the Node window.
   b. Click on the **New** button.
   c. On the resulting dialog, choose to define a port and select the link protocol type.

      CS Linux displays the port configuration dialog.

   d. Enter the basic port parameters as described in “Configuring Connectivity” on page 74.
2. Click on the **Advanced** button at the bottom of the dialog.

   CS Linux displays the Port Parameters dialog. The lower pane shows settings that affect downstream LU templates.
3. Select the *Configure downstream LUs for implicit PU access* option.
4. Click on **OK**.

   CS Linux displays the Downstream LU Template Configuration dialog.
5. Enter appropriate values in the fields on the dialog.
6. Click on **OK** to define the implicit downstream LU template.

Defining Downstream LUs

Before configuring a downstream LU for SNA gateway, you need the following information:

- LU name for each downstream LU. (This is a local identifier, and does not have to match the configuration of the downstream system.)
- LU number for each downstream LU.
- Link station to the downstream node.
- Upstream LU name (for the host LU).

To configure a downstream LU for SNA gateway, perform the following steps:

1. Select the link station to the downstream node in the Connectivity pane of the Node window.
2. Click on the **New** button.
3. Select **New downstream LU** and click on **OK**.

   CS Linux displays the Downstream LU dialog.
4. Enter appropriate values in the fields on the dialog.
5. Click on **OK** to define the downstream LU.

   The LU definition appears in the Connectivity pane of the Node window, below the link station to the downstream node.

Configuring DLUR

In addition to providing direct access to a host computer, CS Linux can provide dependent LU requester (DLUR) facilities. This feature enables sessions for dependent LUs to span multiple nodes in an APPN network, instead of requiring a direct connection to the host.

Normally, a dependent LU session requires a direct communications link to the host computer. If many nodes (including a host node) are connected together in an APPN network, some of them may not have a direct connection to the host, but only an indirect connection through another node. It is not possible to establish dependent LU sessions to the host from LUs in these indirectly connected nodes.

Dependent LU requester (DLUR) is an APPN feature designed to overcome this limitation.

DLUR on an APPN node (such as a CS Linux node) works in conjunction with dependent LU server (DLUS) at the host, to route sessions from dependent LUs on the DLUR node across the APPN network to the DLUS host. The route to the host can span multiple nodes and can take advantage of APPN's network management, dynamic resource location, and route calculation facilities. DLUR must be available on the node where the LUs are located, and DLUS must be available on the host node, but DLUR is not required on any intermediate nodes in the session route.

If the CS Linux DLUR node is a network node or a Branch Network Node, it can also provide passthrough DLUR facilities for dependent LUs on downstream computers connected to the CS Linux node. These LUs can use DLUR on the CS Linux node to access the host across the network, in the same way as for LUs internal to the node. The downstream computers do not run DLUR, and indeed do not need to be aware that DLUR is being used.

Figure 8 shows a CS Linux server configured as an APPN network node, implementing passthrough DLUR to support sessions between LUs on the host (the upstream node) and LUs on the nodes in the APPN network (downstream nodes).

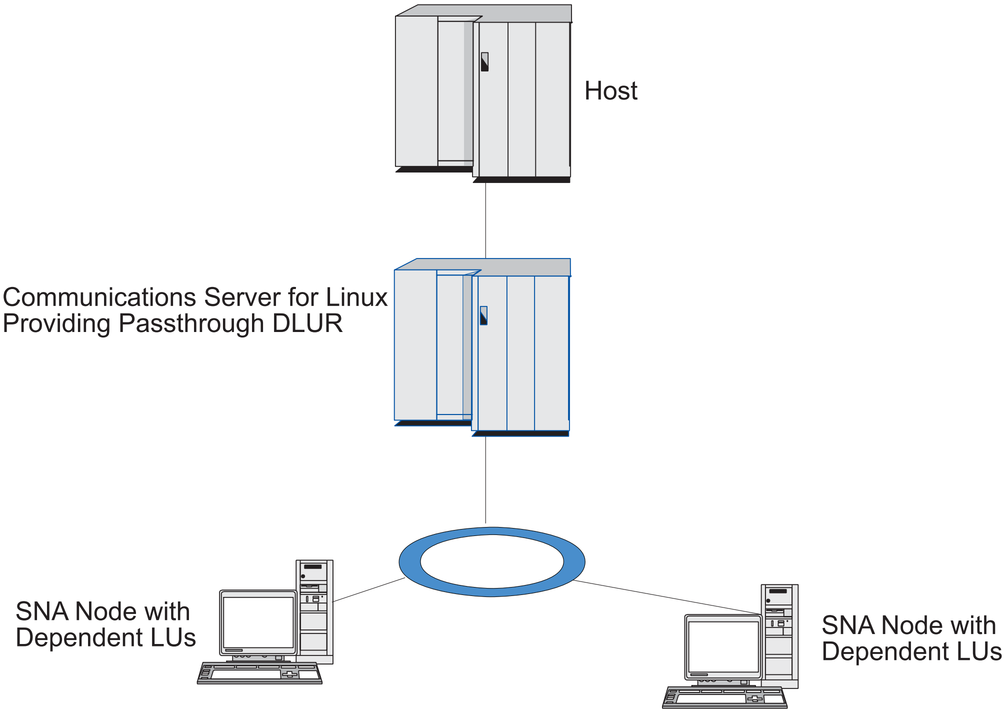


Figure 8. CS Linux Node Providing DLUR

Note:

1. You cannot configure DLUR on a LEN node.
2. You can configure passthrough DLUR only on a network node or a Branch Network Node.
3. If you are using Branch Extender, you cannot configure DLUR on an end node in the branch (with a Branch Network Node as its network node server). However, you can support dependent LU applications from this node by configuring passthrough DLUR on the Branch Network Node (so that the end node in the branch does not run DLUR, but uses passthrough DLUR on the Branch Network Node).

The tasks you need to perform to configure DLUR depend on whether the dependent LUs are on the local node or on downstream nodes.

Configuring DLUR Support on the Local Node

You need the following information for this task:

- PU ID for the PU on the local node.
- PU name. (This is a local identifier, and does not have to match the host configuration.)
- Name of the DLUS on the host (and the name of the backup DLUS if there is one).
- LU name, LU number, and LU type for each downstream LU. The LU number must match the number configured on the host.

To configure DLUR support on the local node, you must perform the following configuration tasks:

1. Define the local node as described in "Configuring the Node" on page 74. If you are providing passthrough DLUR support for downstream nodes, define the node as an APPN network node or branch network node.
2. Configure connectivity to the APPN network. APPN connectivity requires at least a port and link station for independent traffic between the local node and the adjacent APPN network node, as described in "Configuring Connectivity" on page 74.
3. Define a DLUR PU on the local node (the DLUR PU supports connectivity to the host).

   To configure the DLUR PU, perform the following steps from the Node window:

   a. Select the **Services** menu, then the **Connectivity** submenu, then **New DLUR PU** (or click on the **New** button in the button bar, then select **DLUR PU**).

      When you click on the **OK** button, CS Linux displays the DLUR PU Configuration dialog.

   b. Enter appropriate values in the fields on the dialog.

   c. Click on the **OK** button to define the DLUR PU.

      The DLUR PU appears in the Connectivity pane below the DLUR item.
4. To configure DLUR to support LUs on the local node, you must add the LUs on the local node. The LUs must be configured to support LUA, as described in "Configuring LUA" on page 85. Depending on the requirements of the user applications supported by the LUs, you may also need to perform further configuration.

Configuring Passthrough DLUR Support for Downstream Nodes

You need the following information for this task:

- Downstream PU name for each downstream node, or for each PU on the downstream node. (This is a local identifier, and does not have to match the host configuration.)
- Name of the DLUS on the host.

To configure passthrough DLUR support for downstream nodes, you must perform the following configuration tasks:

1. Define the local node as an APPN network node (see "Configuring the Node" on page 74).
2. Configure connectivity to the downstream nodes. Configure ports and link stations for dependent traffic between the local node and each downstream

node, as described in “Configuring Connectivity” on page 74. (You do not need to define a DLUR PU to support passthrough DLUR for downstream nodes.)

3. A downstream node can support multiple PUs. In this case, each downstream PU is associated with a different link, so you need to configure multiple links between the CS Linux DLUR node and the downstream node, and you need to know the downstream PU name for each link.

Configuring TN Server

3270 emulation programs that communicate over TCP/IP (rather than over an SNA network) are referred to as TN3270 programs (Telnet 3270 emulation programs).

TN3270 programs can also include support for TN3270E (Telnet 3270 standard extensions). TN3270E is an open protocol that supports 3270 device emulation (including both terminals and printers) using Telnet. It enables a Telnet client to select a particular device (by specifying the LU name), and provides enhanced support for various functions, including the ATTN and SYSREQ keys and SNA response handling.

Note: This guide uses the term TN3270 for information that applies equally to the TN3270, TN3287, and TN3270E protocols.

CS Linux TN server provides access to 3270 host computers for TN3270 users on other computers. TN server enables TN3270 users to share a host connection with CS Linux or with other TN3270 users, instead of requiring a direct link. TN server also enables TN3270 users to access hosts that are not running TCP/IP.

The CS Linux TN server feature provides an association between a TN3270 user and CS Linux 3270 LU. All data from the TN3270 user is routed to the LU. This means that the configuration for both the host and the TN3270 user is as though they were connected directly; neither needs to be aware that data is being routed through TN server.

CS Linux TN server supports all TN3270 client emulation programs that correctly implement the protocols defined in IETF RFCs 1123, 1576, 1646, 1647, and 2355.

When a TN3270 program communicates with TN server, CS Linux identifies the program by the TCP/IP address of the computer where the TN3270 program is running. CS Linux cannot distinguish between two different TN3270 programs being used by different users on the same computer. In the CS Linux manuals, the term TN server user refers to the computer where a TN3270 program is running, not to an individual user of that program.

Each TN server user connecting to CS Linux using the TN3270 Server feature is normally configured to access a single 3270 LU, and so is restricted to one host session at a time. However, you can also configure a TN server user to access a pool of 3270 LUs, instead of having a single dedicated 3270 LU for each user. This enables users to access as many sessions as there are available LUs in the pool.

Before you can configure TN server access, you must perform the following configuration tasks:

- Define the local node as described in “Configuring the Node” on page 74.
- Configure a port and link station for dependent traffic between the local node and the host, as described in “Configuring Connectivity” on page 74.

To configure TN server access, you must perform the following configuration tasks:

- Define the 3270 LUs on the local node that are used for communication with the host. To add the LUs, see “Defining 3270 LUs.”
- If you are going to use any LU pools, define them as described in “Defining an LU Pool.”

Defining 3270 LUs

Before configuring the 3270 LU, gather the following information:

- LU name. (This is a local identifier, and does not have to match the host configuration.)
- LU number (or numbers for a range of LUs).
- LU type (3270 display model or 3270 printer).
- Pool name (if you are adding the LU to a pool).

To configure an LU of types 0–3 for a previously defined link station, perform the following steps from the Node window:

1. Select the link station to the host in the Connectivity pane of the window.
2. Click on the **New** button.
3. Select the LU type (**New 3270 display LU** or **New 3270 printer LU**) on the resulting dialog.

   When you select this item and click on **OK**, CS Linux displays the LU Type 0–3 dialog.
4. Enter appropriate values in the fields on the dialog.
5. Click on **OK** to define the LU.

   The LU appears in the Connectivity pane of the Node window, below the link station to the host.

Defining an LU Pool

For 3270, you can define LU pools to simplify user configuration and provide greater flexibility in establishing host sessions. For example, you can define several 3270 LUs in a single LU pool, then configure multiple TN3270 clients using this LU pool. This makes configuring the 3270 sessions easier and enables any client to use any LU in the pool.

Note: You can assign a TN3270 client either to a specific LU or to an LU pool.

- If you assign the client to a specific LU that is in a pool, the client uses this LU if it is available; otherwise it uses any free LU from the pool, as though you had assigned it to the LU pool instead of the specific LU.
- If you want the client to use only a specified LU, so that the client's session cannot be established if the LU is already in use, ensure that the LU is not in a pool.

You can view the LU pools for the local CS Linux node using the LU Pools window. This window lists the LU pools configured on the local system, and enables you to select LUs to add to an LU pool.

You can add the following LU types to a pool for use by 3270 (do not mix LUs of different types in the same pool):

- 3270 display LU
- Unrestricted LU

Before you can add LUs to a pool, the LUs must be defined on the local node.

To configure an LU pool, perform the following steps from the Node window:

1. Select **LU Pools** from the **Windows** menu.

   CS Linux displays the LU Pools window.
2. Click on the **New** button.

   CS Linux displays the LU Pool Configuration dialog.

   The box on the right lists LUs that are not yet allocated to any pool. Any of these LUs can be included in the new pool.
3. Select the LU or LUs you wish to add to the pool, and click on the **New** button to move the selected LUs to the box on the left.

   To remove an LU from the box on the left, select it and click on the **Remove** button.
4. Click on **OK** to define the LU pool.

   All of the LUs in the box on the left are added to the LU pool.

   The pool appears in the LU Pools window.

Configuring TN3270 Server

Before configuring TN3270 server, you need the following information:

- Whether the server supports only TN3270, or also TN3270E (which includes TN3270 support).
- Whether a TN3270E client can request a specific LU.
- Display and printer LU names (or LU pool names) for each client. (Printer LU names are needed only if you are supporting TN3270E.)
- If only certain clients are permitted, or if you want to restrict certain clients to specific LUs, you need the TCP/IP name or address of the client.
- TCP/IP port number on the TN server node.
- Whether SSL data encryption, client authentication, and server authentication are required (this option is available only if you have installed the additional software required to support it).

To associate a display LU and printer LU, you also need the names of those LUs. A TN server association record defines an association between a printer LU and display LU, so that the TN3270E protocol can connect the two. You do not need to define an association record if you are not supporting TN3270E or if you are not supporting printer LUs.

The TN server defaults record defines parameters that are used on all TN3270 client sessions. You can define a single defaults record for each server.

To configure TN3270 server, perform the following steps from the Node window:

1. Define a TN server access record:
   a. Select **TN Server** from the **Services** menu.

      CS Linux displays the TN Server window, which lists all the configured TN server access records in the upper pane and TN server association records in the lower pane.

   b. Select the pane that contains TN3270 Server access records and click on the **New** button.

      CS Linux displays the TN Server Access dialog.

   c. Enter appropriate values in the fields on the dialog.

d. Click on **OK** to define the TN server access record. The record appears in the TN Server window.

2. Define a TN server association record:

   a. Select the pane that contains association records in the TN Server window and click on the **New** button.

      CS Linux displays the TN Server Association Record dialog.

   b. Enter appropriate values in the fields on the dialog.

   c. Click on **OK** to define the TN server association record. The record appears in the TN Server window.

3. If you need to force printer responses, specify a keep-alive method for all TN3270 sessions, specify how to access the external LDAP server that holds a revocation list used to check authorization for TN3270 clients, or use TN3270 SLP (Service Location Protocol), use the TN Server Advanced Parameters dialog to do this.

For more information about configuring SSL support for TN Server, refer to the IBM Communications Server Support web pages at http://www.ibm.com/software/network/commserver/support/.

Configuring TN Redirector

The CS Linux TN Redirector feature provides passthrough TCP/IP host access to TN3270, TN3270E, TN5250 and VT clients, referred to collectively as Telnet clients. The Telnet user communicates with CS Linux over a TCP/IP connection; CS Linux then communicates with the host over another TCP/IP connection. This allows you to use Secure Sockets Layer (SSL) security checking where necessary, and not on the complete user-to-host connection. For example:

- If clients are connecting to CS Linux over a TCP/IP LAN where checking is not required, but are connecting to a remote host that requires SSL, you can use SSL over the TCP/IP connection between CS Linux and the host. This means that security is checked once for all clients, and individual clients do not have to provide security information.
- If CS Linux is installed on the same site as the host, but clients are connecting in from external sites, you can use SSL over the client connections to CS Linux without having to install SSL software on the host.

Configuring TN Redirector

Before you can configure TN Redirector access, you must define the local node as described in “Configuring the Node” on page 74. You also need the following information:

- If only certain clients are permitted, you need the TCP/IP name or address of the client.
- TCP/IP port number used by the client to connect to the TN Redirector node.
- TCP/IP name or address of the host.
- TCP/IP port number used by the TN Redirector node to connect to the host.
- Whether SSL data encryption, client authentication, and server authentication are required between the client and the TN Redirector node (this option is available only if you have installed the additional software required to support it).
- Whether SSL data encryption is required between the TN Redirector node and the host.

The TN redirector defaults record defines parameters that are used on all TN redirector client sessions. You can define a single defaults record for each client TCP/IP port number.

To configure TN redirector, perform the following steps from the Node window to define a TN redirector access record:

1. Select **TN Server** from the **Services** menu.

   CS Linux displays the TN Server window, which lists all the configured TN3270 server access records, TN3270 server association records, and TN Redirector access records.
2. Select the pane that contains TN Redirector access records and click on the **New** button.

   CS Linux displays the TN Redirector Access dialog.
3. Enter appropriate values in the fields on the dialog.
4. Click on **OK** to define the TN Redirector access record. The record appears in the TN Redirector pane of the TN Server window.

Note: The SNA node must be active in order to use TN Redirector, although it does not use any of the node's SNA resources.

Disabling CS Linux

Disabling the CS Linux software automatically stops the CS Linux node and its associated connectivity components. Disabling CS Linux also stops any other processes (such as an LUA application) from using CS Linux resources on this server.

In general, you should stop individual services as users finish using them, and only disable the system when there is no CS Linux activity.

If you need to disable CS Linux while users are active, warn users that CS Linux is stopping, and give them time to finish their activities before you disable the software.

When you disable the CS Linux software, applications using the APPC, CSV, LUA, NOF, or MS APIs are notified by a `COMM_SUBSYSTEM_ABENDED` return code and CPI-C applications by a `CM_PRODUCT_SPECIFIC_ERROR` return code.

To disable the CS Linux software, enter the following command at the Linux command prompt:

```
sna stop
```

If CS Linux is disabled successfully, **sna stop** returns an exit code of `0`. Any other exit code indicates that an error occurred and that the CS Linux software was not disabled. Refer to *IBM Communications Server for Data Center Deployment on Linux Diagnostics Guide* for more information about exit code values.

Starting CS Linux Automatically

Bringing up CS Linux to fully working status can be viewed as several steps:

- Enabling the CS Linux software
- Initializing the SNA node
- Activating the ports and link stations configured on the node.

Each of these steps can be managed separately, and all steps can be done at boot time if necessary. The rest of this section describes these steps and explains how to control them at boot time.

Enabling CS Linux

CS Linux requires several kernel modules to be loaded in order to operate. These modules are loaded when you first enable the CS Linux software, and are unloaded only when you change the kernel run level.

CS Linux also requires a number of daemons (programs) to be running. Until these daemons are running, you cannot configure or use CS Linux.

- To start the daemons and enable the CS Linux software, issue the command **sna start**.
- To stop the daemons and disable the CS Linux software, issue the command **sna stop**.

Initializing the SNA node

When the CS Linux software has been enabled, you can configure the SNA node and its resources, typically using the Motif administration program xsnaadmin. However, the node is not available for use until it is initialized. You can initialize it from the command-line administration program, using the command **snaadmin init_node**, or from the Motif administration program xsnaadmin.

Activating ports and link stations

Ports and link stations can be configured to start in different circumstances:

- By operator intervention only
- On demand (when an application starts that uses a resource on the link)
- On node start-up (when the command **snaadmin init_node** is issued, or when the node is started from the Motif administration program).

Starting by operator is the default, but you can change this for a particular port or link by using the command-line administration program or the Motif administration program.

Note: Starting a port enables that port to receive calls from other computers, but does not allow it to make outgoing calls. Starting a link station means that CS Linux attempts to contact the remote computer.

Starting CS Linux at reboot time

In common with other Linux services, CS Linux is enabled at reboot. In other words, by default after a reboot, the command **sna start** is issued but the SNA node is not started.

This initialization is done in the CS Linux boot time initialization script, **/etc/rc.d/init.d/snastart**. As is conventional for start-up scripts, this is linked to **/etc/rc?.d/init.d/snastart** for the various boot levels.

You can edit **/etc/rc.d/init.d/snastart** to change what happens at reboot. The most common change is to add the initialization of the node. The command for this, **snaadmin init_node**, is already included in the file but commented out, so you simply need to uncomment it. By including this command, you also trigger the activation of any ports or link stations that are configured to be activated on node start-up.

Starting CS Linux Automatically

Applications that use CS Linux should not be started until after the node has been initialized. If required, you can start these applications automatically at boot time, and add any other **snaadmin** commands that you need to run at boot time, in one of two ways:

- Add commands at the end of **/etc/rc.d/init.d/snastart**, after the **snaadmin init_node** command.
- Create an **/etc/rc?.d/init.d** script with a number greater than `95`, which ensures that it will run after CS Linux is started, and add the commands to that script.

Note: Changes that you make to the **/etc/rc.d/init.d/snastart** file will not be saved when you upgrade CS Linux to a later version. Always ensure that you keep a copy of your changes, so that you can re-apply them after an upgrade.

Chapter 9. Information Resources for CS Linux and SNA

This chapter describes the resources in the SNA library that provide information about SNA technology and the many networking products and services that IBM offers. It also describes information that is available in network forums.

SNA Library

The SNA library includes marketing brochures, books, user guides, and tutorials that provide both introductory and in-depth information about the following topics:

- SNA theory
- SNA products
- Product implementation
- Configuration of systems and networks
- SNA application programs and APIs
- Overall planning, performance, and tuning
- Problem diagnosis
- Network management
- Network security

All IBM publications can be ordered through your IBM representative, the IBM branch serving your locality, or by calling IBM directly at 1-800-879-2755.

For a list of publications most relevant to CS Linux, see the Bibliography at the end of this book.

For information about additional publications, contact your IBM representative.

Network-Accessible Information

To promote information exchange, IBM sponsors electronic forums and bulletin boards. It posts home pages on the Internet and provides online documentation that is also accessible on the World Wide Web.

Product Support over IBMLink

The IBMLink forum is held over IBM-owned networks. It is designed to help customers with licensed IBM products solve technical problems and other issues related to their system and network. IBM personnel answer questions and mediate online discussions among IBM customers.

For more information about IBMLink, use http://www.ibmlink.ibm.com.

Information in IBM Home Pages

On the Internet, various IBM home pages provide access to forums. For comprehensive help, the IBM main home page can be used to navigate to information centers on the Internet and the World Wide Web. The main home page can be accessed by using http://www.ibm.com.

You can access information about IBM networking software, including CS Linux, by using http://www.ibm.com/software/network. Information about CS Linux is at http://www.ibm.com/software/network/commserver.

For more detailed information about support for CS Linux, use http://www.ibm.com/software/network/commserver/support.

Information for Downloading

On the World Wide Web, users can download Redbook publications by using http://www.redbooks.ibm.com.

Information on IBM software can be accessed at http://www.ibm.com/software, where you can link to pages about CS Linux and all of the IBM Software Servers.

Suggested Reading

For those who want to strengthen their understanding of SNA, the following books cover SNA theory and the use of CS Linux in practice. The books are helpful to both novices and experts who might need either a starting point for getting acquainted with SNA or an in-depth treatment of the subject.

- *Systems Network Architecture: Technical Overview* (GC30–3073)
- *IBM Communications Server for Data Center Deployment on Linux Administration Guide*

If you have more specific interests, see the Bibliography for other references, or contact your local IBM representative.

Appendix. Notices

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

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

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

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

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

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

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

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

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

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

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

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

Trademarks

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

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

Bibliography

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

- CS Linux, Version 7.0
- Systems Network Architecture (SNA)
- Host configuration
- z/OS Communications Server
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- X.25
- Advanced Program-to-Program Communication (APPC)
- Programming
- Other IBM networking topics

For books in the CS Linux library, brief descriptions are provided. For other books, only the titles and order numbers are shown here.

CS Linux Version 7.0 Publications

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

- *IBM Communications Server for Data Center Deployment on Linux Quick Beginnings* (GC31-6768 and GC31-6769)

  This book is a general introduction to CS Linux, including information about supported network characteristics, installation, configuration, and operation. There are two versions of this book:

  GC31-6768 is for CS Linux on the i686, x86_64, and ppc64 platforms
  GC31-6769 is for CS Linux for System z.
- *IBM Communications Server for Data Center Deployment on Linux Administration Guide* (SC31-6771)

  This book provides an SNA and CS Linux overview and information about CS Linux configuration and operation.
- *IBM Communications Server for Data Center Deployment on Linux Administration Command Reference* (SC31-6770)

  This book provides information about SNA and CS Linux commands.
- *IBM Communications Server for Data Center Deployment on AIX or Linux CPI-C Programmer's Guide* (SC23-8591)

  This book provides information for experienced "C" or Java programmers about writing SNA transaction programs using the CS Linux CPI Communications API.
- *IBM Communications Server for Data Center Deployment on AIX or Linux APPC Programmer's Guide* (SC23-8592)

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

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

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

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

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

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

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

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

  This book provides a comprehensive list of terms and definitions used throughout the CS Linux library.

Systems Network Architecture (SNA) Publications

The following books contain information about SNA networks:

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

Host Configuration Publications

The following books contain information about host configuration:

- *ES/9000, ES/3090 IOCP User's Guide Volume A04* (GC38-0097)
- *3174 Establishment Controller Installation Guide* (GG24-3061)
- *3270 Information Display System 3174 Establishment Controller: Planning Guide* (GA27-3918)
- *OS/390 Hardware Configuration Definition (HCD) User's Guide* (SC28-1848)

z/OS Communications Server Publications

The following books contain information about z/OS Communications Server:

- *z/OS V1R7 Communications Server: SNA Network Implementation Guide* (SC31-8777)
- *z/OS V1R7 Communications Server: SNA Diagnostics* (Vol 1: GC31-6850, Vol 2: GC31-6851)
- *z/OS V1R6 Communications Server: Resource Definition Reference* (SC31-8778)

TCP/IP Publications

The following books contain information about the Transmission Control Protocol/Internet Protocol (TCP/IP) network protocol:

- *z/OS V1R7 Communications Server: IP Configuration Guide* (SC31-8775)
- *z/OS V1R7 Communications Server: IP Configuration Reference* (SC31-8776)
- *z/VM V5R1 TCP/IP Planning and Customization* (SC24-6125)

X.25 Publications

The following books contain information about the X.25 network protocol:

- *Communications Server for OS/2 Version 4 X.25 Programming* (SC31-8150)

APPC Publications

The following books contain information about Advanced Program-to-Program Communication (APPC):

- *APPC Application Suite V1 User's Guide* (SC31-6532)
- *APPC Application Suite V1 Administration* (SC31-6533)
- *APPC Application Suite V1 Programming* (SC31-6534)
- *APPC Application Suite V1 Online Product Library* (SK2T-2680)
- *APPC Application Suite Licensed Program Specifications* (GC31-6535)
- *z/OS V1R2.0 Communications Server: APPC Application Suite User's Guide* (SC31-8809)

Programming Publications

The following books contain information about programming:

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

Other IBM Networking Publications

The following books contain information about other topics related to CS Linux:

- *SDLC Concepts* (GA27-3093)
- *Local Area Network Concepts and Products: LAN Architecture* (SG24-4753)
- *Local Area Network Concepts and Products: LAN Adapters, Hubs and ATM* (SG24-4754)
- *Local Area Network Concepts and Products: Routers and Gateways* (SG24-4755)
- *Local Area Network Concepts and Products: LAN Operating Systems and Management* (SG24-4756)
- *IBM Network Control Program Resource Definition Guide* (SC30-3349)

Index

Communicating your comments to IBM

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

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

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

Send your comments to us in any of the following ways:

- To send comments by FAX, use this number: 1+919-254-1258
- To send comments electronically, use this address: comsvrcf@us.ibm.com
- To send comments by post, use this address:

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

Make sure to include the following in your note:

- Title and publication number of this document
- Page number or topic to which your comment applies.

Product Number: 5725-H32

Printed in USA

GC31-6769-04