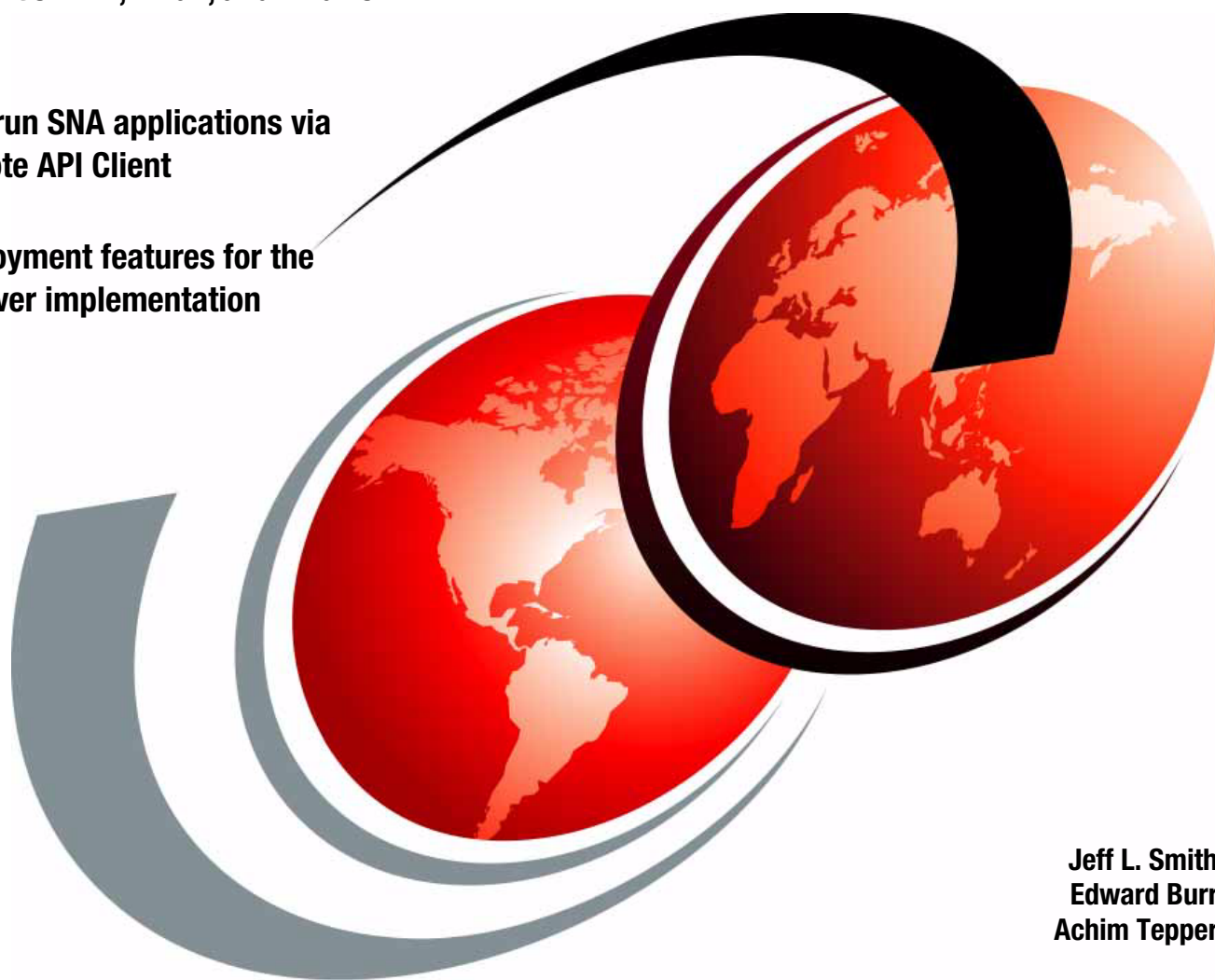


IBM Communications Server for Data Center Deployment V7.0

Deployment on AIX, Linux, and Linux on
System z

Ability to run SNA applications via
IBM Remote API Client

New deployment features for the
client/server implementation



Jeff L. Smith
Edward Burr
Achim Tepper



International Technical Support Organization

**IBM Communications Server for Data Center
Deployment V7.0**

September 2013

Note: Before using this information and the product it supports, read the information in “Notices” on page v.

First Edition (September 2013)

This edition applies to IBM Communications Server for Data Center Deployment V7.0.

© Copyright International Business Machines Corporation 2013. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	vii
Now you can become a published author, too!	viii
Comments welcome	viii
Stay connected to IBM Redbooks	viii
Chapter 1. Communications Server for Data Center Deployment overview	1
1.1 Communications Server for Linux	2
1.2 Remote API client/server	2
1.3 New features	2
1.4 Web Administration package	2
Chapter 2. IBM Communications Server for Linux	3
2.1 Installation	4
2.1.1 Upgrading from previous levels of CS Linux	4
2.1.2 Preparing for CS Linux installation	4
2.1.3 Changing the language environment variable	4
2.1.4 Installing prerequisite software	5
2.1.5 Examples	5
2.1.6 Installing the CS Linux licensed program	7
2.2 Configuring CS Linux	9
2.3 Diagnosing CS Linux	10
2.3.1 Log files	11
2.3.2 Tracing	11
Chapter 3. Remote API client/server	13
3.1 Windows client	15
3.1.1 Installing the client	15
3.1.2 Configuring the client	16
3.1.3 Diagnosing the client	19
3.2 Linux client	20
3.2.1 Installing the client	20
3.2.2 Configuring the client	21
3.2.3 Diagnosing the client	22
3.3 AIX client	23
3.3.1 Installing the client	23
3.3.2 Configuring the client	24
3.3.3 Diagnosing the client	24
Chapter 4. New features	27
4.1 TN3270 device mapping table	28
4.2 Virtualization support and high-performance routing	30
4.3 LU alias mapping	31
4.4 Tracking LU6.2 resources	32
4.5 Remote API Client and IBM Personal Communications	33

Chapter 5. Web Administration package	35
5.1 Installing WebAdmin	37
5.2 Configuring WebAdmin	37
5.3 Administering WebAdmin	40
5.3.1 Node management	40
Related publications	45
IBM Redbooks	45
Online resources	45
How to get Redbooks	46
Help from IBM	46

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 grant 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.

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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

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.

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 illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®	Redbooks®	System p®
IBM®	Redpaper™	System z®
Rational®	Redbooks (logo)  ®	VTAM®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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.

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

Preface

This IBM® Redpaper™ publication will guide the user through the installation, configuration, and administration of IBM Communications Server for Data Center Deployment V7.0. It is not intended to be all-inclusive. Rather, it builds on previous publications referenced throughout the document. The focus is on the consolidation of IBM Systems Network Architecture (SNA) resources, key features and functions available in IBM Communications Server for Data Center Deployment, and the Web Administration package specific to the Linux platform.

Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Raleigh Center.



Jeff L. Smith is the Chief Programmer for the development of the Distributed Communications Servers in Enterprise Networks solution for Application and Integration Middleware (AIM), Software Group, IBM. He has worked at IBM for 24 years in various roles supporting multiprotocol networking software products. Jeff has a Master of Science degree in Computer Engineering from Florida Atlantic University, and a Bachelor of Science degree in Physics from Appalachian State University.



Edward Burr is a Communications Server Software Analyst at Research Triangle Park (RTP), North Carolina. Edward has worked for IBM for 14 years in various roles supporting IBM Communications Server on IBM AIX®, Linux, and Microsoft Windows platforms. Edward holds a Bachelor of Science degree in Science Education from the University of Oklahoma.



Achim Tepper is an Advisory IT Specialist in Germany. He has 30 years of experience in client support with IBM, and is a member of the support team for networking products for the two European (formerly Europe, Middle East, and Africa, or EMEA) integrated operating teams (IOTs), focusing on distributed networking support. His areas of expertise include IBM Communications Server for AIX, IBM Communications Server for Linux, IBM Rational® Host On-Demand, and IBM Personal Communications. He was one of the coauthors of the IBM Redbooks® publication, *Host Access Client Package V4 Update*, SG24-6182.

Thanks to the following people for their contributions to this project:

Joshua Bennetone, Joe Welsh, and Octavio L. Ferreira
IBM RTP and IBM Brazil

Rufus P. Credle Jr.
ITSO, Raleigh Center

Thanks to the authors of the previous editions of this paper.

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our papers to be as helpful as possible. Send us your comments about this paper or other IBM Redbooks publications in one of the following ways:

- ▶ Use the online **Contact us** review Redbooks form found at:

ibm.com/redbooks

- ▶ Send your comments in an email to:

redbooks@us.ibm.com

- ▶ Mail your comments to:

IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- ▶ Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- ▶ Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- ▶ Look for us on LinkedIn:
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with Rich Site Summary (RSS) Feeds:
<http://www.redbooks.ibm.com/rss.html>



Communications Server for Data Center Deployment overview

IBM Communications Server for Data Center Deployment V7.0 is the follow-on release to Communications Server for AIX V6.4, Communications Server for Linux V6.4 (CS Linux), and Communications Server for Linux on IBM System z® V6.4. The new Communications Server for Data Center Deployment combines a variety of functionalities into one product offering:

- ▶ Server support for AIX, Linux, and Linux on System z
- ▶ Client/server support for Microsoft Windows, AIX, and Linux

This allows IBM Systems Network Architecture (SNA) to be configured and managed in a more central location, reducing the cost of administration and concentrating SNA resources.

The goal of the data center deployment model is to provide a cloud-like implementation, in which the SNA resources are shared across servers to provide redundancy and high availability.

This approach is described in *SNA Modernization Strategy for Access Node Connectivity*, REDP-4871:

<http://www.redbooks.ibm.com/abstracts/redp4871.html?open>

1.1 Communications Server for Linux

The Communications Server for AIX is described in *IBM Communications Server for AIX, V6: New Features and Implementation Scenarios*, SG24-5947:

<http://www.redbooks.ibm.com/abstracts/sg245947.html?Open>

The differences between Communications Server for AIX and CS Linux are minor. The directory paths and a few messages are different, but they are 99% the same product. This section provides installation, configuration, and diagnostic information to help you implement a CS Linux domain for the remote application programming interface (API) clients.

1.2 Remote API client/server

The remote API client (IBM Remote API Client) that is included with the Communications Server for Data Center Deployment provides a thin SNA client. This client connects over Transmission Control Protocol/Internet Protocol (TCP/IP) to a domain of SNA servers on AIX or Linux. This section provides installation, configuration, and administration examples of an implementation of the client/server, covering Windows, AIX, and Linux platforms.

1.3 New features

The Communications Server for Data Center Deployment provides some newly documented features that help with the deployment aspect of a client/server implementation. There are new features that address issues where the client is mobile, or appears to be mobile, in an SNA network:

- ▶ The ability to map logical unit (LU) aliases from clients to aliases on servers
- ▶ The ability to track the LU6.2 SNA resources on the clients using LU aliases

Additional examples of the following functionality are provided:

- ▶ TN3270 device-support table
- ▶ TN3270 Server hardware encryption support
- ▶ Improved default values for a progressive-mode adaptive rate-based (ARB) flow control algorithm (Progressive ARB, used in Enterprise Extender, or EE, connections)

1.4 Web Administration package

The implementation of the Communications Server for Data Center Deployment on Linux, and especially on Linux for System z, requires the ability to manage the server remotely. There is no native hardware monitor on Linux for System z. Also, the operation of the X Window System graphical user interface (GUI) can affect the network performance if there are many SNA connections displayed.

The Web Administration (WebAdmin) package supplies a secure, browser-enabled management interface for the server. It provides basic command-line interface (CLI) queries and Perl script outputs. These give you status about, and some management of, the node and link resources. This section provides the installation and configuration examples that help you implement WebAdmin access to CS Linux.



IBM Communications Server for Linux

The IBM Communications Server for Linux (CS Linux) and the Communications Server for Linux on System z are mainly the same product. An exception is that Linux on System z supports a channel-to-channel multi-path channel (CTCMPC) connection, but the Linux version on Intel and IBM System p® does not.

In this section, we describe how to install, configure, and diagnose CS Linux. The Communications Server for AIX is much the same code. There are a few exceptions:

- ▶ Some directory paths
- ▶ Some installation steps
- ▶ Some established function parameters that AIX contains that Linux does not

Whenever the CS Linux function is described in this paper, the Communications Server for AIX version will either be the same, or there will be a slight difference in platform-specific implementation (for instance, directory paths might be different).

2.1 Installation

This section covers the installation steps for CS Linux and Communications Server for Linux on System z.

2.1.1 Upgrading from previous levels of CS Linux

If you are planning to upgrade from a previous level of CS Linux, see the chapter on migrating in *IBM Communications Server for Data Center Deployment on Linux, Quick Beginnings*.

2.1.2 Preparing for CS Linux installation

Before you start with the installation, make sure that your system meets the hardware and software requirements explained in the section on installation requirements in *IBM Communications Server for Data Center Deployment on Linux, Quick Beginnings*.

For Intel and System p platforms, Linux supports a power management feature. This can disable some hardware functions (spin down drivers, some CPUs, and so on) to save power on system resources. CS Linux, however, requires the hardware to be active full time.

Before installing CS Linux, you need to check whether your Linux machine includes some form of Advanced Power Management (APM) features. If it does, you should ensure that these features are disabled. As an example, to remove the APM feature from your PC, use the following command:

```
rpm -e apmd
```

2.1.3 Changing the language environment variable

When you use CS Linux, make sure that the language (**LANG**) environment variable is set correctly, to indicate the language that you want to use. Use the following command to change the **LANG** variable:

```
export LANG=<language>
```

Replace *<language>* with the identifier for the language that you want to use, which can be one of the following options (shown in Table 2-1).

Table 2-1 Language options

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

Identifier	Language
zh_CN	Chinese (Simplified EUC)
zh_TW	Chinese (Traditional)

2.1.4 Installing prerequisite software

Before installing CS Linux, you need to install the prerequisite software:

- ▶ Location information service (LiS) Linux Streams
- ▶ OpenMotif
- ▶ Java
- ▶ Secure Sockets Layer (SSL)

For full details of the software packages required, and step-by-step instructions for installing them, see the readme file on the CS Linux CD. Note that some packages are 32-bit and some are 64-bit, and both packages are sometimes needed on 64-bit systems.

Pay special attention to the installation of the prerequisite open source package, Linux Streams, described in 2.1.5, “Examples”. Follow the instructions to install and build the LiS package.

2.1.5 Examples

The following screen captures show an example of typical steps to install CS Linux:

1. Install the Linux Streams LiS-2.19.0 package, and then apply the LiS-2.19.0-CS7000 patch file, as shown in Figure 2-1.

```

root@dicsr2/usr/src/LiS-2.19
[root@dicsr2 src]# tar -xzf LiS-2.19.0.tgz
[root@dicsr2 src]#
[root@dicsr2 src]#
[root@dicsr2 src]# cd LiS-2.19
[root@dicsr2 LiS-2.19]# patch -g0 -p1 < /mnt/cslinux/patches/LiS-2.19.0-CS7000.patch
patching file buildLiS
patching file config.mk
patching file Configure
patching file drivers/str/Config
patching file drivers/str/fifo.c
patching file drivers/str/linux/connld.c
patching file drivers/str/linux/ldl.c
patching file drivers/str/linux/Makefile
patching file drivers/str/linux/mtdrv.c
patching file drivers/str/loop.c
patching file drivers/str/mk.bdy
patching file drivers/str/pipemod.c
patching file drivers/str/relay.c
patching file drivers/str/sad.c
patching file head/head.c
patching file head/linux/exports.c
patching file head/linux/lislocks.c
patching file head/linux/lismem.c
patching file head/linux/lispci.c

```

Figure 2-1 Linux Streams LiS-2.19.0 installation

2. Change the permissions for the newly created `buildLiS` script command, and then execute the script. It will build the LiS package automatically (see Figure 2-2).

```

root@dicsr2:/usr/src/LiS-2.19
[ root@dicsr2 LiS-2.19 ]#
[ root@dicsr2 LiS-2.19 ]# chmod +x buildLiS
[ root@dicsr2 LiS-2.19 ]# ./buildLiS
LiS Configure script version 2.100 10/11/04 invoked from main makefile

CONFIG=/usr/src/LiS-2.19/config.in

package=LiS
prefix=/usr
execprefix=/usr
includedir=/usr/include
libdir=/usr/lib
bindir=/usr/bin
sbindir=/usr/sbin
sysconfdir=/etc
datadir=/usr/share
pkgincludedir=/usr/include/LiS
pkglibdir=/usr/lib/LiS
pkgdatadir=/usr/share/LiS
pkgsrkdir=/usr/src/LiS
mandir=/usr/share/man
modsubdir=misc

KBIN is ///usr/src/kernels/2.6.32-358.2.1.el6.i686
Will use ///usr/src/kernels/2.6.32-358.2.1.el6.i686/include for header files.
Kernel support for SMP detected.
No kernel support for mod-versions detected.

```

Figure 2-2 Change permission window

3. After the installation is finished, you should see the following message, as shown in Figure 2-3:

BuildLiS done, no errors reported

```

root@dicsr2:/usr/src/LiS-2.19
LD [M] /usr/src/LiS/km26/streams-relay3.ko.unsigned
NO SIGN [M] /usr/src/LiS/km26/streams-relay3.ko
CC /usr/src/LiS/km26/streams-timod.mod.o
LD [M] /usr/src/LiS/km26/streams-timod.ko.unsigned
NO SIGN [M] /usr/src/LiS/km26/streams-timod.ko
CC /usr/src/LiS/km26/streams-tirdwr.mod.o
LD [M] /usr/src/LiS/km26/streams-tirdwr.ko.unsigned
NO SIGN [M] /usr/src/LiS/km26/streams-tirdwr.ko
CC /usr/src/LiS/km26/streams.mod.o
LD [M] /usr/src/LiS/km26/streams.ko.unsigned
NO SIGN [M] /usr/src/LiS/km26/streams.ko
make: Leaving directory `/usr/src/kernels/2.6.32-358.2.1.el6.i686'
../util/linux/makenodes
cp -p ../util/linux/makenodes /usr/sbin/strmakenodes
cp -p ../util/linux/streams /usr/sbin/streams
mkdir -p /lib/modules/`uname -r`/misc
cp -p streams*.ko /lib/modules/`uname -r`/misc
mv /etc/modprobe.conf.streams /etc/modprobe.d/streams.conf
chmod 755 ../scripts/clean-modprobe.conf
../scripts/clean-modprobe.conf /etc/modprobe.conf
../scripts/clean-modprobe.conf /etc/modprobe.conf.local
depmod
BuildLiS done, no errors reported
[ root@dicsr2 LiS-2.19 ]#

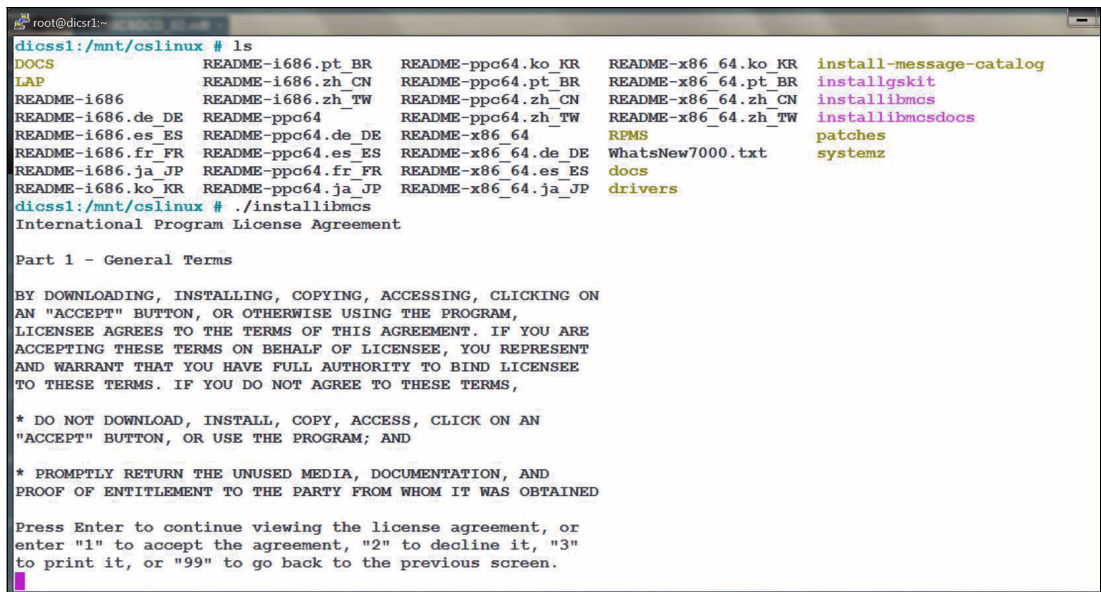
```

Figure 2-3 BuildLiS done window

2.1.6 Installing the CS Linux licensed program

After you have installed the prerequisite software, as described in 2.1.4, “Installing prerequisite software” on page 5, you are ready to install CS Linux. Use the following procedure:

1. Log in with root privileges.
2. Run the `installibmcs` program from the CS Linux CD. See the readme file on the same source for details of the commands required to accomplish this. The program normally prompts you to provide the following information:
 - Confirm that you have read and accepted the CS Linux license terms (Figure 2-4).



```
root@dicsr1:~# cd /mnt/cslinux
dicsr1:/mnt/cslinux # ls
DOCS          README-i686.pt_BR  README-ppc64.ko_KR  README-x86_64.ko_KR  install-message-catalog
LAP           README-i686.zh_CN  README-ppc64.pt_BR  README-x86_64.pt_BR  installgskit
README-i686   README-i686.zh_TW  README-ppc64.zh_CN  README-x86_64.zh_CN  installibmcs
README-i686.de_DE  README-ppc64      README-ppc64.zh_TW  README-x86_64.zh_TW  installibmcsdocs
README-i686.es_ES  README-ppc64.de_DE  README-x86_64      RPMS                  patches
README-i686.fr_FR  README-ppc64.es_ES  README-x86_64.de_DE  WhatsNew7000.txt     systemz
README-i686.ja_JP  README-ppc64.fr_FR  README-x86_64.es_ES  docs
README-i686.ko_KR  README-ppc64.ja_JP  README-x86_64.ja_JP  drivers
dicsr1:/mnt/cslinux # ./installibmcs
International Program License Agreement

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON
AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM,
LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE
ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT
AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE
TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN
"ACCEPT" BUTTON, OR USE THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND
PROOF OF ENTITLEMENT TO THE PARTY FROM WHOM IT WAS OBTAINED

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, or "99" to go back to the previous screen.
```

Figure 2-4 License agreement window

- Specify the name of the server that will be the master server in the CS domain.
If you want to run CS Linux as a stand-alone node, do not specify this parameter (just press Enter).

If you do not specify the master server, the node will not support client/server functions (see Figure 2-5, and the Notes box following step 3).

```

root@dics1:~
Program.

"Program" - the following, including the original and all
whole or partial copies: 1) machine-readable instructions
and data, 2) components, files, and modules, 3) audio-
visual content (such as images, text, recordings, or
pictures), and 4) related licensed materials (such as keys
and documentation).

"Proof of Entitlement" ("PoE") - evidence of Licensee's
Authorized Use. The PoE is also evidence of Licensee's
eligibility for warranty, future update prices, if any, and
potential special or promotional opportunities. If IBM does

Press Enter to continue viewing the license agreement, or
enter "1" to accept the agreement, "2" to decline it, "3"
to print it, or "99" to go back to the previous screen.
1
Installing Communications Server for Linux...
Preparing... ##### [100%]
  1:ibm-commserver ##### [100%]
Installed driver for kernel 3.0.42-0.7-pae_Oct_9_11:58:45_UTC_2012_(a8dc443) OK
Enter master server name, <ENTER> for stand alone system

Preparing... ##### [100%]
  1:gskcrypt32 ##### [100%]
Preparing... ##### [100%]
  1:gskssl32 ##### [100%]
installgskit is done
installibmcs is done
dics1:/mnt/cslinux #

```

Figure 2-5 CS Linux install window

- If you need to perform 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>
```

In this example, `<master_name>` is the name of the master server. Again, if you want to run CS Linux as a stand-alone node, do not specify this parameter, and the node will not support client/server functions.

Notes:

- ▶ Regardless of whether you perform an attended or unattended installation, the installation process does not allow you to configure the domain name. Therefore, the default domain name of `ibmcs_domain` will be used. If you do not want to use the default domain name, do not specify a master server at this time. Instead, use the `snanetutil` command, as described in 2.2, "Configuring CS Linux" on page 9, to set the master server and domain name.
- ▶ Installing CS Linux automatically reconfigures the IBM Systems Network Architecture (SNA) device drivers on the local node. If you are installing on a computer with limited memory (for instance, only 64 MB), you might need to reboot after installing CS Linux before the SNA node can be started.
- ▶ You can install adapters after CS Linux is installed. You do not need to reinstall CS Linux after adding an adapter.

4. Add the CS Linux binary directories to your `PATH` environment variable. You might want to change your profile to add these 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
```

5. If you are running 64-bit applications, change the environment variable:

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

6. Start CS Linux. After installation, this will happen automatically when the machine is rebooted:

```
cd /  
sna start
```

7. If you installed the IBM Global Security Kit (GSKit) software to use Telnet (TN) Server or TN Redirector with the SSL feature, you need to configure the SSL software after you have installed CS Linux.

For more details, see the information about configuring SSL for use with TN Server or TN Redirector in *IBM Communications Server for Data Center Deployment on Linux, Quick Beginnings*.

8. If you want to install the CS Linux Portable Document Format (PDF) documentation, run the `installibmcsdocs` program from the CS Linux CD.

9. If the server will have firewall settings active, the ports for Enterprise Extender (EE), TN3270 Server, and the remote application programming interface (API) client/server (IBM Remote API Client) will need to be defined and enabled.

For more details, see the information about configuring your firewall to allow EE, client/server, and TN3270 Server traffic in *IBM Communications Server for Data Center Deployment on Linux, Quick Beginnings*.

2.2 Configuring CS Linux

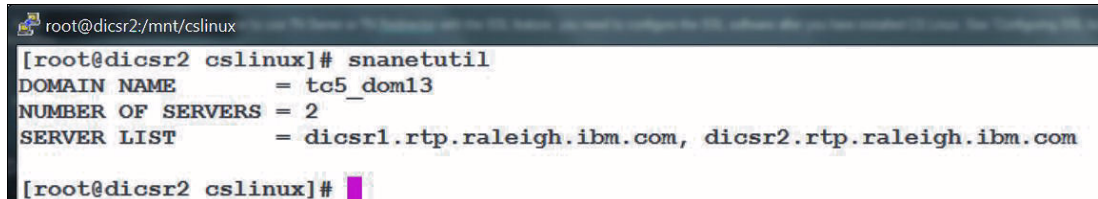
Before the server can be started, if the server will provide support for Remote API Client, the *domain name* and the *master server* for the domain must first be defined. This step can be skipped if there is no Remote API Client support for this configuration. Follow this procedure to configure CS Linux:

1. If you want to set up a domain that is unique (not one with the default domain name), or if you skipped the process of specifying a master server earlier in the installation process, you can now configure the master server and the domain name using this command:

```
snanetutil <server-name> <domain-name>
```

In this example, `<server-name>` is the name of your master server, and `<domain-name>` is the unique domain name.

2. The `snanetutil` command without any parameters displays the current settings, as shown in Figure 2-6.



```
root@dicsr2:/mnt/cslinux
[root@dicsr2 cslinux]# snanetutil
DOMAIN NAME      = tc5_dom13
NUMBER OF SERVERS = 2
SERVER LIST      = dicsr1.rtp.raleigh.ibm.com, dicsr2.rtp.raleigh.ibm.com
[root@dicsr2 cslinux]#
```

Figure 2-6 The `snanetutil` settings

3. CS Linux starts with an empty configuration. If an X Window System GUI is available, you should use the `xnaadmin` GUI application as referenced in *IBM Communications Server for AIX, V6: New Features and Implementation Scenarios*, SG24-5947. If a command line is all that is available, use the following `snaadmin` commands to define the initial SNA node configuration:

```
snaadmin define_node, cp_alias=server1, fqcp_name=appn.server1
snaadmin define_ip_dlc, dlc_name=IPDLC
snaadmin define_ip_port, port_name=IPPORT, dlc_name=IPDLC
snaadmin define_ip_ls, ls_name=LSPORT, port_name=IPPORT, initially_active =
YES, remote_ip_host=10.10.18.11
```

However, if this is not possible, you can minimally define a node as an end node (default), so that it will at least start and connect to a partner node.

4. To configure each server other than the master server as a backup server, enter the following command:

```
snaadmin add_backup, backup_name=<server_name>
```

In this example, `<server_name>` is the name of the server that you want to add as a backup server.

You can run this command on the backup server, or on the master server, but the CS Linux software must be running on the master server in either case.

5. Server information is stored in the `/etc/sna/sna.net` file on AIX, and in the `/etc/opt/ibm/sna/sna.net` file on Linux.

2.3 Diagnosing CS Linux

In addition to the `snanetutil` command described in 2.2, “Configuring CS Linux” on page 9, the output of the following `snaadmin` commands provides additional information about the environment and its current status:

- The `snaadmin query_sna_net` command returns information about servers that can act as backup master servers, as defined in the `/etc/opt/ibm/sna/sna.net` file. This command can be used to obtain information about a specific server, or about multiple servers, depending on the options used.

The order of server names in this file is significant. The first server listed in the file will always be the master if it is active, the second will be the master if the first is inactive, the third will be the master if the first and second are both inactive, and so on. Because of this ordering, the list of server names returned using the `snaadmin query_sna_net` command is in the same order as it is in the `sna.net` file.

- ▶ The `snaadmin query_node_all` command returns information about nodes in the CS Linux domain. The command returns only a list of node names, and does not provide detailed information about the node's configuration. To obtain detailed information about a particular node (*<node name>*), you can use the `snaadmin query_node_<node name>` command.
- ▶ The `snaadmin query_rapi_clients` command shows the Domain Name System (DNS) and Internet Protocol (IP) address of AIX, Linux, or Windows clients for which the CS Linux server is currently acting as the master. It enables the administrator to see which clients are attached to the server, and which logical units (LUs) are in use.
- ▶ The `netstat -a` command verifies that a Transmission Control Protocol (TCP) port and the User Domain Protocol (UDP) port 1553 show a status of LINUX.

2.3.1 Log files

Communications Server for AIX and CS Linux log errors in the `sna.err` file, found in the `/var/sna` directory for AIX, and the `/var/opt/ibm/sna` directory for Linux:

1. To see the last 40 lines of the file, use the `tail` command:

```
tail -40 /var/opt/ibm/sna/sna.err
```

It is useful to enable the audit logging initially, so that you can follow the activation and deactivation of links, sessions, and applications.

2. If the system starts to get heavily used, it might be better to disable audit logging to provide better performance for the transactions.
3. On Communications Server for AIX, the audit log is in the `/var/sna/sna.aud` file. For CS Linux, the audit log is in the `/var/opt/ibm/sna/sna.aud` file. Audit log messages with time stamps are appended to the end of the file.

2.3.2 Tracing

In case of a problem, you might be advised by support personnel to take traces:

1. Start the traces with the following commands:

```
snaadmin set_log_type,audit=YES  
snaadmin  
set_trace_file,trace_file_type=CS,trace_file_size=1000000,file_name=filename
```

The default file size is one file with 10 MB. The default file name is `/var/sna/snacs1.trc` on AIX, and `/var/opt/ibm/sna/snacs1.trc` on Linux.

```
snaadmin set_trace_type,trace_flags=SLIM+APPC+NOF+LUA
```

This is a minimum. You can also just specify ALL to get every trace point.

```
snaadmin set_cs_trace,trace_flags=ALL  
snaadmin add_dlc_trace
```

This will record the link in addition to the client/server connection.

2. Re-create the problem, or wait until it recurs, and then stop the traces:

```
snaadmin set_trace_type,trace_flags=NONE
```

3. In addition, run either of the following commands:

```
snagetpd -q /tmp/filename
```

or

```
snaadmin remove_dlc_trace
```




Remote API client/server

The remote application programming interface (API) client/server (IBM Remote API Client) enables a client system to run IBM Systems Network Architecture (SNA) applications without having a complete SNA stack. It connects across a Transmission Control Protocol/Internet Protocol (TCP/IP) network to a domain of servers that has one or more IBM Communications Server for Linux (CS Linux), CS Linux on System z, or Communications Server for AIX servers.

The Remote API Client can be installed on seven platforms.

The Microsoft Windows client has been tested with the following Windows operating system versions:

- ▶ The Windows 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, or 32-bit Windows 8.
- ▶ The Windows 64-bit 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, Windows Server 2008 R2, Windows 7 x64, or Windows 8 x64.

The Linux client has been tested with the following Linux operating system versions. It might also run satisfactorily on other Linux distributions:

- ▶ Red Hat Enterprise Linux 5 (RHEL5)
- ▶ Red Hat Enterprise Linux 6 (RHEL6)
- ▶ SUSE Linux Enterprise Server 10 (SLES10)
- ▶ SUSE Linux Enterprise Server 11 (SLES11)

The Linux client for System z has been tested with the following Linux operating system versions. It might also run satisfactorily on other Linux distributions:

- ▶ Red Hat Enterprise Linux 5 for System z (RHEL5-s390x)
- ▶ Red Hat Enterprise Linux 6 for System z (RHEL6-s390x)
- ▶ SUSE Linux Enterprise Server 10 for IBM mainframes (SLES10-s390x)
- ▶ SUSE Linux Enterprise Server 11 for IBM mainframes (SLES11-s390x)

The AIX client has been tested with the following AIX operating system versions:

- ▶ AIX 6.1 or later
- ▶ AIX 7.1 or later

3.1 Windows client

This section provides information about the installation of IBM Remote API Client on Windows.

3.1.1 Installing the client

The installation files for Remote API Client on Windows are self-extracting installation programs. They execute the following actions:

- ▶ Prompt for the acceptance of the software license.
- ▶ Install the code.
- ▶ Invoke the configuration GUI.
- ▶ Provide a choice to display the readme file when the installation is completed.

To perform the installation, follow these steps:

1. Install the `i_w32cli.exe` file or the `i_w64cli.exe` file from the CD on your client PC. After the installation has finished, the installation process will automatically start the `sxc1conf` configuration tool (see 3.1.2, “Configuring the client” on page 16).
2. Remote API Client on Windows runs as a Windows service. The installation program configures it to start automatically when the computer starts. If necessary, you can start it manually in either of the following ways:
 - Start the client service from the Services applet, Remote API Client, under **Control Panel** → **Administrative Tools**.
 - Type `net start sxc1ient` from a command window, or from **Start** → **Run**.
3. Before disabling the client, ensure that all CS Linux applications on the client have been stopped (3270 and 5250 emulation programs, or applications using the CS Linux APIs). To disable the client, stop the Client service in one of the following ways:
 - Stop the Client service from the Services applet under **Control Panel** → **Administrative Tools**.
 - Type `net stop sxc1ient` from a command window, or from **Start** → **Run**.

On a computer running Windows Terminal Services, this prevents all users from using the client.

Silent installation

The IBM Remote API Client on Windows installation package can be extracted into a temporary directory. You can then start the `setup.exe` file with parameters to configure the domain, servers, and other connection parameters in one step.

For details about the parameter settings for the command-line installation, see the section on installing IBM Remote API Client software from the command line in the *IBM Communications Server for Data Center Deployment on Linux, Quick Beginnings* manual.

The following script starts an unattended install of Remote API Client for Windows on a remote system, as shown in Example 3-1.

Example 3-1 Start an unattended install

```
REM Script to install Remote API Client on Windows
REM
REM Change the '-i' parameter to match the domain in your network,
REM      the '-u' parameters to match the hostname of the servers,
```

```

REM          the '-j' parameter to specify the Reconnect timeout for your client
REM
setup -id313_domain -udicsr2.rtp.raliegh.ibm.com -udicsr1.rtp.raleigh.ibm.com -b0
-j20 -f2C:\instrapi.log -y -accept -s
REM Display services to see if service exists
net start
REM
REM If the service does not exist, uncomment the following 2 lines to create
REM   and start the service
REM
REM sc create SxCClient binPath= C:\IBMCS\w32cli\sxc1svce.exe DisplayName= "IBM
Remote API Client" start= auto depend= lanmanworkstation
REM sc start SxCClient
REM
REM
REM end

```

3.1.2 Configuring the client

To configure Remote API Client, follow these steps:

1. From a command prompt, change to the directory where the client software is installed, and then run the **sxc1conf** command. A window similar to that shown in Figure 3-1 should open.

To start the configuration tool in a specific language, use the **sxc1app1** command.

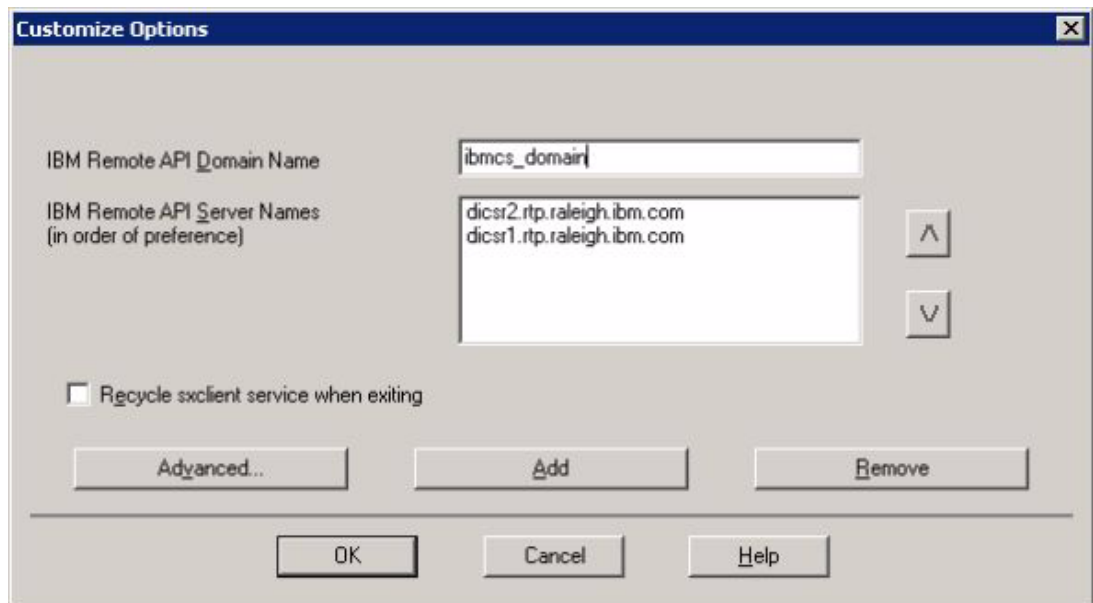


Figure 3-1 The Customize Options window

2. In the registry, the information is contained in values configured under subkeys of the following key:

```

\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA Client\SxCClient\Parameters

```

The configuration data from the Customize Options window shown in Figure 3-1 on page 16 appears in the registry as shown in Example 3-2.

Example 3-2 Configuration data

```
Key 'Configuration'  
"invoked_tps"="yes"  
"client_start_timeout"=dword:0000000a  
"snagroup"=""  
"domain"="ibmcs_domain"  
"lan_access_timeout"=""  
"server_lost_timeout"="20"  
"broadcast_attempt_count"="5"
```

```
Key 'Servers'  
"Server1"="dicsr2.rtp.raleigh.ibm.com"  
"Server2"="dicsr1.rtp.raleigh.ibm.com"
```

3. The Client Monitor places an icon in the system tray that displays the client's status when you move the mouse pointer over it. The Client Monitor shows one of the following statuses:
 - Not Active. The client has not been started.
 - Not Connected. The client has been started, but has not yet made contact with a server (or has lost contact).
 - *<Server_Name>*. The client is connected to the named server.
4. The Client Monitor is set up to run automatically when the computer starts, but you can also run it manually in either of the following ways:
 - From the **Start** menu, choose **Client Monitor** from the Remote API Client on Windows program group.
 - From a command prompt, change to the directory where the client software is installed, and run the **sxc1app1** command.
5. The Client Monitor normally displays information in the language that you selected when you installed the client software. If you prefer to see this information in a different language, you can do so by starting the Client Monitor from a command prompt:
 - a. Change to the directory where the client software is installed, and then to the subdirectory corresponding to your preferred language.
 - b. Run the **sxc1app1** command. For example, to see the information in French, run the **sxc1app1** command from the fr_FR subdirectory below the directory where the client software is installed.

Invoking a transaction program

Based on the operating system, you can specify (in different ways) the applications that get dynamically started when a remote partner sends an ATTACH. The server will associate the local logical unit (LU) to a client, based on the **sys_name** parameter for the LU.

Configuring the local LU and a transaction program

There are several ways to specify the local LU alias in Remote API Client on Windows:

- ▶ Using the **APPCLLU** environment variable
- ▶ In the registry
- ▶ In the Common Programming Interface for Communications (CPI-C) side information table

In order, Remote API Client checks the environment variable first, and uses this alias if it is specified. If the alias is not specified in the environment variable, it uses the registry entry. If both the environment variable and the registry entry are not defined, the CPI-C side information record is used to specify the LU.

You might need to use the environment variable if you are using Windows Terminal Server, and you need to run multiple copies of the same application using different local LUs. The definition of a transaction program (TP), which is invoked by a call from a partner LU, is in the Windows registry. The registry key is in the following format:

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA Client\SxCClient\Parameters\<MyExeName>
```

In this case, *<MyExeName>* is the file name of the program without the .exe extension.

The **APPCLU** and **APPCTPN** environment variable values under this registry key specify the local LU alias and the executable file.

For instance, to have the myApp application set to use LU ATCLU01, use a script, such as the script shown in Example 3-3.

Example 3-3 MyApp setting

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA Client\SxCClient\Parameters
myApp
APPCTPN = myApp.exe
APPCLU = ATCLU01
```

This is also a solution if the application does not propagate a definite LU or an LU alias, and therefore is using the control program (CP) LUname of the server. Instead of setting the registry values, you can also use the following environment variables:

```
APPCTPN="myApp.exe"
APPCLU=ATCLU01
```

If the Remote API Client runs in a Terminal Server environment, this setting might also help. Each Terminal Server has one Remote API Client, but each client that logs in should have the **APPCLU** environment variable set to the LU alias for the LU assigned to that user.

For instance, the user John might have the LU NETA.JOHN1, which has the LU alias of john. The environment variable for John's login session on the Terminal Server should be **APPCLU=john** (LU alias is case-sensitive).

The Remote API Client provides a sample set of TPs called the Advanced Program-to-Program Communication (APPC) Application Suite. The *IBM Communications Server for Linux V7.0 APPC Application Suite User's Guide*, SC23-8595, provides details for the sample ASuite applications provided. The sample file is installed in `\ibmcs\w32cli\asuite.tps` (Win32) or `\ibmcs\w64cli\asuite.tps` (Win-x64).

The TP definitions can be set using the **tpinst32** command-line program. See the appendixes in *IBM Communications Server for Data Center Deployment on Linux, Administration Guide* for full details about the parameters. The **tpinst32** program sets all of the Windows registry keys as needed, and as provided by the .tps file.

3.1.3 Diagnosing the client

In the registry, tracing control information is contained in values configured under sub-keys of the following key (see Example 3-4).

Example 3-4 Value configured under sub-keys

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA Client\SxClient\Parameters
```

```
Key 'API_tracing' (11 values, 0 subkeys)
  file1      REG_SZ      : 'C:\IBMCS\w64cli\snaapi1.trc'
  file2      REG_SZ      : 'C:\IBMCS\w64cli\snaapi2.trc'
  cpic       REG_SZ      : 'yes'
  ms         REG_SZ      : 'no'
  rui        REG_SZ      : 'yes'
  nof        REG_SZ      : 'yes'
  csv        REG_SZ      : 'yes'
  appc       REG_SZ      : 'yes'
  all_api    REG_SZ      : 'yes'
  truncation_length REG_DWORD : 0x0000ffff (65535)
  flip_size  REG_DWORD : 0x000f4240 (1000000)
```

```
Key 'CS_tracing' (8 values, 0 subkeys)
  file1      REG_SZ      : 'C:\IBMCS\w64cli\snacs1.trc'
  file2      REG_SZ      : 'C:\IBMCS\w64cli\snacs2.trc'
  send       REG_SZ      : 'yes'
  receive    REG_SZ      : 'yes'
  admin_msg  REG_SZ      : 'yes'
  datagram   REG_SZ      : 'yes'
  data       REG_SZ      : 'yes'
  flip_size  REG_DWORD : 0x000f4240 (1000000)
```

Log files (sna.err, sna.aud, and sna_log.msg) are saved in the C:\IBMCS\w32cli or C:\IBMCS\w64cli installation directory.

The **snagetpd** utility collects all of the Windows trace files and other Windows Client system information, and gathers them into a single self-extracting compressed file named snapd.exe in the installation directory.

Tracing

Either turn the trace on and off in the GUI by selecting **Sxclconf** → **Advanced** (as shown in Figure 3-1 on page 16) or by editing the registry keys.

In Example 3-5, scripts change the related registry entries to either start or stop the trace.

Example 3-5 Scripts change

Start Tracing:

```
@ECHO OFF
REM
REM Set trace file1 and file2 to C:\ibmcs\w32cli\API_APPCx.trc, where x is file
number
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v file1 /f /d
"c:\ibmcs\w32cli\API_APPC1.trc"
```

```

REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v file2 /f /d
"c:\ibmcs\w32cli\API_APPC2.trc"
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v all_api /f
/d "YES"
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v appc /f /d
"YES"
REM

Stop Tracing:

@ECHO OFF
REM
REM UnSet trace file1 and file2
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v file1 /f /d
""
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v file2 /f /d
""
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v all_api /f
/d "no"
REM
reg add "HKLM\SOFTWARE\SNA Client\SxClient\Parameters\API_tracing" /v appc /f /d
"no"
REM

```

3.2 Linux client

This section covers the installation of the Remote API Client on Linux.

3.2.1 Installing the client

To install the Remote API Client on Linux, perform the following steps:

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


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

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

```
cd ibm-commsserver-clients/linux
./installibmccli
```


The shell script will test for certain prerequisites and issue warning messages if they are not met. You will be prompted to read and accept the license agreement, after which the script will install the Red Hat Package Managers (RPMs).

5. Create the client network data file to specify the Communications Server for AIX or CS Linux servers to which the client can connect.

You might also want to set up additional configurations to enable advanced support for tracking LUs and mapping Local LU aliases. This enables you to integrate the clients more dynamically, and to manage the LU resources in a domain of servers. For more details, see the section on managing Remote API Clients in *IBM Communications Server for Data Center Deployment on Linux, Administration Guide*.

6. Add the IBM Remote API Client binary directories to your **PATH** variable. You might want to change your profile to add these automatically (see Example 3-6).

Example 3-6 Profile automatic settings

```
export PATH="$PATH:/opt/ibm/sna/bin"
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
If you are running 64-bit applications, you would use:
export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64
export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64
For Java CPI-C applications you should also set the following environment
variable:
export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar
For some applications you might 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
```

Linux on System z

The installation is almost the same as the installation on Linux, with the following differences:

1. Copy, or use File Transfer Protocol (FTP) to copy, the `ibm-commserver-client-7.0.0.0-s390x.tgz` file from the `/ibm-commserver-clients/linux-systemz` directory on the installation CD to the Linux on System z system. Ensure that you use binary mode to copy the file.
2. Extract the tape archive (`.tar`) file into an empty temporary directory:

```
mkdir /tmp/ibmcs
cd /tmp/ibmcs
tar -xzf ibm-commserver-client-7.0.0.0-s390x.tgz
```

3. From here, follow the same steps as in 3.2.1, "Installing the client" on page 20.

3.2.2 Configuring the client

The client information for a Remote API Client on Linux is stored in the `/etc/opt/ibm/sna/sna_clnt.net` file, which is *not* created automatically.

Before you can start the client, you need to create the file and add some mandatory information. To do so, follow these steps:

1. Using any editor, create the `/etc/opt/ibm/sna/sna_clnt.net` file, and add at least the name of your domain in the following format:

```
domain=ibmcs_domain
```

2. Add one or more server names:

```
dicsr2.rtp.raleigh.ibm.com  
dicsr1.rtp.raleigh.ibm.com
```

3. Add the following line:

```
server_lost_timeout=60
```

An example of a Linux client configuration is shown in Figure 3-2.

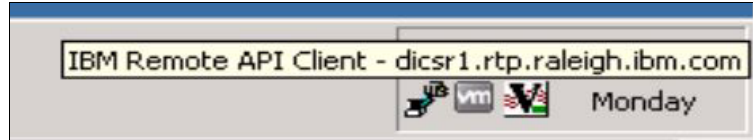


Figure 3-2 Linux client configuration

4. Start the IBM Remote API Client:

```
cd /  
sna start
```

After installation, the Remote API Client on Linux is started automatically when the machine is rebooted.

5. You can check the configuration using the `snaadmin query_sna_net` command.

Configuration of the local LU and a CPI-C transaction program

On Linux and Linux on System z clients, the local LU alias and the CPI-C transaction program can be specified using the `APPCLLU` and `APPCTPN` environment variables:

```
export APPCLLU=ATCLU01  
export APPCTPN=MYTP
```

The TP definitions can be set using the `snatpinstall` command-line program. See the appendixes of the *IBM Communications Server for Data Center Deployment on Linux, Administration Guide* for full details of the parameters. The *IBM Communications Server for Linux V6.2 APPC Application Suite User's Guide*, SC31-6772-00, provides details for the sample ASuite applications provided. The `snatpinstall` command configures the settings as needed to invoke TPs dynamically, called by partner LU applications.

3.2.3 Diagnosing the client

The following commands provide additional information about the environment and its current status:

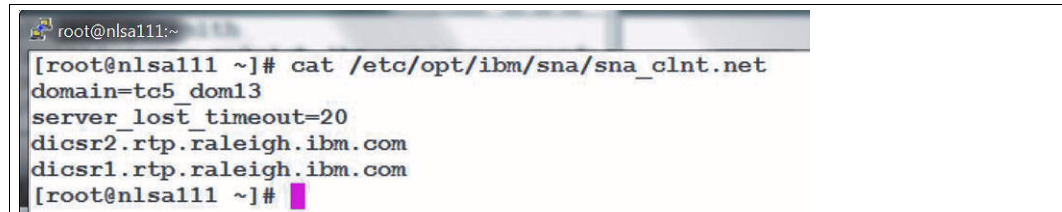
- ▶ To check whether your client is connected, and to which server, run the following command:

```
cat /var/opt/ibm/sna/server.current
```

- ▶ The `snaadmin query_sna_net` command returns information about servers that can act as backup master servers, as defined in the `sna_clnt.net` file. This command can be used to obtain information about a specific server, or about multiple servers, depending on the options that you use.

The order of server names in this file is significant. The first server listed in the file will always be the master if it is active, the second will be the master if the first is inactive, the third will be the master if the first and second are both inactive, and so on.

Because of this ordering, the list of server names returned using `snaadmin query_sna_net` is in the same order as it is in the file, as shown in Figure 3-3.



```
root@nlsall1:~  
[root@nlsall1 ~]# cat /etc/opt/ibm/sna/sna_clnt.net  
domain=tc5_dom13  
server_lost_timeout=20  
dicsr2.rtp.raleigh.ibm.com  
dicsr1.rtp.raleigh.ibm.com  
[root@nlsall1 ~]#
```

Figure 3-3 Query_sna_net window

- ▶ The `snaadmin query_node_all` command returns information about nodes in the CS Linux domain. The command returns only a list of node names and does not provide detailed information about the node's configuration.

The files required for any diagnostics (`sna.err`, `sna.aud`, and `sna_log.msg`) are saved in the `/var/opt/ibm/sna/` directory.

- ▶ The `snagetpd` utility collects all of the trace files and other client system information, and gathers them into a single compressed file named `pd.tar.gz` in the current directory.

Tracing

You might be requested by support personal to start and stop a trace:

1. Start the trace using the following command:

```
sna start -t
```

2. Re-create the problem or wait until it recurs:

```
sna stop
```

3.3 AIX client

This section covers the installation of the IBM Remote API Client on AIX.

3.3.1 Installing the client

To install the Remote API Client on AIX, perform the following steps:

1. Copy (or use FTP to copy) the `sna.client.7.0.0.0.I` file from the `/ibm-commserver-clients/aix` directory on the installation CD to the AIX workstation. Ensure that you use binary mode to copy the file.
2. Log in to the AIX workstation as `root`.
3. Install the AIX Client using either `smit` or `installp`. For instructions about how to install the client, 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 Communications Server for AIX servers to which the client can connect.

You might also want to set up additional configurations to enable advanced support for tracking LUs and mapping Local LU aliases. This enables you to integrate the clients more

dynamically, and to manage the LU resources in a domain of servers. See the section on managing Remote API Clients in the *IBM Communications Server for Data Center Deployment on AIX Administration Guide Version 7.0*, SC31-8586, for more details.

3.3.2 Configuring the client

The client information for a Remote API Client on AIX is stored in the `/etc/sna/sna_clnt.net` file, which is *not* created automatically.

Before you can start the client, you need to create the file, and add some mandatory information:

1. Using any editor, create the `/etc/sna/sna_clnt.net` file, and add at least the name of your domain in the following format:

```
domain=ibmcs_domain
```

2. Add one or more server names:

```
dicsr2.rtp.raleigh.ibm.com  
dicsr1.rtp.raleigh.ibm.com
```

3. Add the following line:

```
server_lost_timeout=60
```

4. Start the IBM Remote API Client:

```
cd /  
  
sna start
```

After installation, the Remote API Client on AIX is started automatically when the machine is rebooted.

5. You can check the configuration using the `snaadmin query_sna_net` command.

Configuring the local LU and a CPI-C transaction program

On AIX clients, the local LU alias and the CPI-C transaction program can be specified using the `APPCLLU` and `APPCTPN` environment variables:

```
export APPCLLU=ATCLU01  
export APPCTPN=MYTP
```

The TP definitions can be set using the `snainstall` command-line program. See the Administration Guide for full details of the parameters. The *IBM Communications Server for Data Center Deployment on AIX or Linux APPC Application Suite User's Guide Version 7.0*, SC23-8595, provides details for the sample ASuite applications provided. The `snainstall` command will configure the settings as needed to invoke TPs dynamically, called by partner LU applications.

3.3.3 Diagnosing the client

The following commands provide additional information about the environment and its current status:

- ▶ To check whether your client is connected to a server, and to which server, run the following command:

```
cat /var/sna/server.current
```

- ▶ The `snaadmin query_sna_net` command returns information about servers that can act as backup master servers, as defined in the `sna_clnt.net` file. This command can be used to

obtain information about a specific server, or about multiple servers, depending on the options used.

The order of server names in this file is significant. The first server listed in the file will always be the master if it is active, the second will be the master if the first is inactive, the third will be the master if the first and second are both inactive, and so on.

Because of this ordering, the list of server names returned using `snaadmin query_sna_net` is in the same order as it is in the file, as shown in Figure 3-4.

```
root@dicsr2:/mnt/cslinux
[root@dicsr2 cslinux]# snaadmin query_sna_net

-----
list_options = FIRST_IN_LIST
domain_name = tc5_dom13
security = OFF

server_name = dicsr1.rtp.raleigh.ibm.com

server_name = dicsr2.rtp.raleigh.ibm.com

-----
[root@dicsr2 cslinux]#
```

Figure 3-4 Query_sna_net window

- ▶ The `snaadmin query_node_all` command returns information about nodes in the AIX domain. The command returns only a list of node names, and does not provide detailed information about the node's configuration.

The files required for any diagnostics (`sna.err`, `sna.aud`, and `sna_log.msg`) will be saved in the `/var/sna/` directory.

- ▶ The `snagetpd` utility collects all of the trace files and other client system information, and gathers them into a single compressed file named `pd.tar.gz` in the current directory.

Tracing

You might be requested by support personal to start and stop a trace:

1. Start a trace:

```
sna start -t
```

2. Re-create the problem or wait until it recurs:

```
sna stop
```




New features

Communications Server for Data Center Deployment provides some newly documented features that help in the deployment aspect of a client/server implementation.

4.1 TN3270 device mapping table

When a TN3270 client connects to the host through either IBM Communications Server for AIX Telnet (TN) Server or Communications Server for Linux (CS Linux) TN Server using Dynamic Definition of Dependent Logical Units (DDDLU), information needs to be sent to the host about the logical unit (LU) model required by the client.

The host normally determines the LU model using a standard mapping from the terminal type (device type) specified by the client. This information can also be found in *IBM Communications Server for Data Center Deployment on Linux, Administration Guide*.

If you need to change the mapping between TN3270 device types and LU models, you can use the `tn3270dev.dat` file. A sample version of this file is provided in `/usr/lib/sna/samples` on AIX, and `/opt/ibm/sna/samples` on Linux.

Copy this file into `/etc/sna` on AIX or `/etc/opt/ibm/sna` on Linux, and make your changes there using a standard ASCII text editor, such as `vi`. The changes will take effect the next time that you restart Communications Server.

Each line in the file represents the mapping between a TN3270 device and the LU model string that should be sent to the host for this device. It consists of the following entries, separated by spaces:

- ▶ The first entry is the single character Y if this mapping is for a TN3270E client (using the TN3270 extensions), or N if it is for a standard TN3270 client.
- ▶ The second entry is the terminal type (device type) specified by the client:
 - This is a text string with a maximum of 40 characters.
 - The valid characters are the letters A - Z, numbers zero through nine (0 - 9), the forward slash (/), the “at” sign (@), and the hyphen (-).
- ▶ The third entry is the 7-character LU model string that CS Linux sends to the host to identify the correct LU model for this client.

The provided sample file includes mappings for 18 standard device types in both TN3270 and TN3270E versions (36 entries in all).

You might need to modify this information:

- ▶ If you need to *specify a different LU model string* for one or more of these standard devices, change the third entry on the appropriate line of the file to specify the required 7-character string.
- ▶ If you need to *support additional device types* that are not included in the sample file, add a new line to the file for each device type, using the same format just described (Y or N followed by two string entries).

Example 4-1 shows the provided sample file (`/usr/lib/sna/samples/tn3270dev.dat` for AIX and `/opt/ibm/sna/samples/tn3270dev.dat` for Linux).

Example 4-1 Sample file

```
// IBM CS TN3270 Server device mapping table.
//
// This file allows the content of the machine type and machine model number
// that is sent to the mainframe for dynamically created LUs (DDDLU) to be
// altered from the defaults. These fields comprise a 7 character string, here
// coded in ASCII but sent to the mainframe in EBCDIC, that is matched to the
// VTAM LUGROUP definitions.
```



```

//
// Each line in this file either starts with // to indicate a comment or is
// a mapping between a tn3270 device type and the string sent to the host which
// comprises three ASCII words.
// The first word is in column 1 and should be the character 'Y' or 'N' to
// indicate if tn3270e protocols are being used (DO TN3270E, WILL TN3270E have
// been exchanged with the client).
// The second word starts in column 3 and should be the device type (upper or
// lower case).
// The third word starts after a blank delimiter and should be the 7 character
// string to send to the mainframe. Words longer than 7 characters are truncated
// and words shorter than 7 characters are null padded.
//
// TN3270 TERMINALS
// This first section refers to clients that are not using the tn3270e
// extensions to the protocol, i.e. the "DO TN3270E, WILL TN3270E" has NOT
// been exchanged between the client and server.
//
N IBM-3287-1 3270DSC
N IBM-DYNAMIC 327000@
N IBM-3278-2 3270002
N IBM-3278-3 3270003
N IBM-3278-4 3270004
N IBM-3278-5 3270005
N IBM-3279-2 3270002
N IBM-3279-3 3270003
N IBM-3279-4 3270004
N IBM-3279-5 3270005
N IBM-3278-2-E 3270002
N IBM-3278-3-E 3270003
N IBM-3278-4-E 3270004
N IBM-3278-5-E 3270005
N IBM-3279-2-E 3270002
N IBM-3279-3-E 3270003
N IBM-3279-4-E 3270004
N IBM-3279-5-E 3270005
//
// TN3270E TERMINALS
// This second section refers to clients that are using the tn3270e extensions
// to the protocol, i.e. the "DO TN3270E, WILL TN3270E" has been exchanged
// between the client and server.
//
Y IBM-3287-1 3270DSC
Y IBM-DYNAMIC 327000@
Y IBM-3278-2 3270002
Y IBM-3278-3 3270003
Y IBM-3278-4 3270004
Y IBM-3278-5 3270005
Y IBM-3279-2 3270002
Y IBM-3279-3 3270003
Y IBM-3279-4 3270004
Y IBM-3279-5 3270005
Y IBM-3278-2-E 3270002
Y IBM-3278-3-E 3270003
Y IBM-3278-4-E 3270004

```

Y IBM-3278-5-E 3270005
 Y IBM-3279-2-E 3270002
 Y IBM-3279-3-E 3270003
 Y IBM-3279-4-E 3270004
 Y IBM-3279-5-E 3270005

4.2 Virtualization support and high-performance routing

Hardware improvements continue to enable computers to run faster each year. Virtualization is now a viable way to run servers and clients to optimize power, space, and resources. The Communications Server for Data Center Deployment provides a wider range of parameters than were available in previous product versions.

These parameters enable the Advanced Peer-to-Peer Networking (APPN) and High-Performance Routing (HPR) connections using Enterprise Extender (EE) to react faster and perform better in a virtualized environment.

The progressive-mode adaptive rate-based (Progressive ARB) algorithm provides a way to measure responses without the use of a hardware clock. This is important on a virtual system, because the clock can be virtual. To optimize connections to adjacent IBM VTAM® resources, it is important to make sure that the VTAM-switched, major node definition for the EE connection uses the HPREEARB=PROGRESS on the following nodes:

- ▶ The physical unit (PU) in a switched major node
- ▶ The GROUP connection network in an external communication adapter (XCA) major node
- ▶ The EE model PU in a model major node

The distributed Communications Server defaults to trying to negotiate for the Progressive ARB, and will fall back to Responsive ARB if it is not available. Responsive ARB tries to use the hardware clock to determine transmission control.

The Rapid Transport Protocol (RTP) timers used to respond to status requests and responses for Internet Protocol (IP)-connected sessions (EE) have defaults and ranges that provide better recovery when packet loss is detected. The round-trip timer (RTT) and the short-request timer (SRT) have been initialized to lower values for the start of RTP sessions, so that responses can react faster as data transmission algorithms begin to detect overflow or lost packets (see Table 4-1 for details).

Table 4-1 Timer values

Product	Re-FIFO cap range in milliseconds (ms)	SRT cap range (ms)	RTT default (ms)	SRT default (ms)
Communications Server for AIX/Linux/Linux on System z v6.4	250 - 12,000	500 - 24,000	250	500
Communications Server for Data Center Deployment v7.0	5 - 12,000	5 - 24,000	10	250

4.3 LU alias mapping

By using the LU alias mapping, we enable an LU6.2 application to use the same LU alias across multiple IBM Remote API Clients to map to specific LU aliases on CS Linux or Communications Server for AIX server.

LU alias mapping uses a new configuration file, `lualiasmap.txt`, in one of these directories:

- ▶ The installation directory for a Windows client (for example, `c:\ibmcs\w64cli`)
- ▶ The configuration directory for a UNIX client (`/etc/sna` on AIX, or `/etc/opt/ibm/sna` for Linux)

This file is parsed as follows: Up to 32 text lines can be present in the file. Each line has the following format:

```
<nnnnnnnn> <mmmmmmmm>
```

In this case, `<nnnnnnnn>` is the eight-character ASCII alias (padded with spaces if necessary) provided by the Advanced Program-to-Program Communication (APPC) application, or the Common Programming Interface for Communications (CPI-C) application, in `TP_STARTED`. Also, `<mmmmmmmm>` is the `lu_alias` that is mapped to a local LU alias on the server.

There is precisely one space between the aliases (the mapped alias must start at column 10). Either alias can be all blank (to refer to the default LU pool or control point LU). Lines shorter than 17 characters are padded with blanks. Characters after position 17 are ignored, and can be used for comments. Lines beyond 32 are ignored.

Each `TP_STARTED` issued by the application is matched against the configured list of alias maps until a match is found. When `TP_STARTED` results from the `cmal1c` command, it is also matched.

When an application starts, it tries to use the LU with an alias of MYLU.

If you define a local LU at the server with an alias of MYLU, it will work correctly, but it can be used with only one Remote API Client at a time.

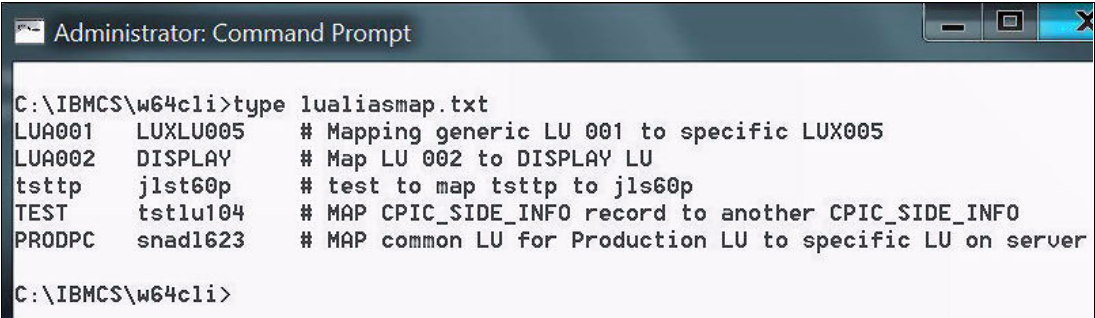
Alternatively, you can set up a unique LU for any Remote API Client:

1. Define `lualiasmap.txt` on each client in the following way:

```
MYLUbbbb ATCLU02
```

2. Define a local LU ATCLU02 at the server.

Figure 4-1 is an example of the `\ibmcs\w64cli\lualiasmap.txt` file.



```
Administrator: Command Prompt
C:\IBMCS\w64cli>type lualiasmap.txt
LUA001  LUXLU005  # Mapping generic LU 001 to specific LUX005
LUA002  DISPLAY  # Map LU 002 to DISPLAY LU
tsttp   jlst60p  # test to map tsttp to jls60p
TEST    tstlu104  # MAP CPIC_SIDE_INFO record to another CPIC_SIDE_INFO
PRODPC  snad1623  # MAP common LU for Production LU to specific LU on server

C:\IBMCS\w64cli>
```

Figure 4-1 `lualiasmap.txt` file

4.4 Tracking LU6.2 resources

The ability to track LU6.2 resources is a new feature of the Communications Server for Data Center Deployment V7.0. This feature is documented in *IBM Communications Server for Data Center Deployment on Linux, Administration Guide*.

This function requires both the server and the client to be configured to support this feature. It is primarily used to connect host applications to IBM Systems Network Architecture (SNA) clients that might not have static Domain Name System (DNS) client names, or to clients that appear to be mobile. The clients might actually not be mobile, but in using terminal servers, they appear to be mobile.

For instance, if the client is running on a Windows Terminal Server farm, the actual Windows Remote API Client for the user might not be the same from day to day. This function enables the domain of distributed communications servers to find the current `sys_name` parameter of the client dynamically.

A program temporary fix (PTF) flag definition is required to configure the server to dynamically update the `sys_name` parameter for a Local LU6.2 definition with the client's IP host name. The new PTF flag option is in the following format:

CLI_OVERWRITE_SYS_NAME

To use the enhancement, follow these steps:

1. Query the existing setting for `ptf_flags`:

```
snaadmin query_node | grep ptf_flag
```

If the response shows a setting (for example, `ptf_flags = SUPPRESS_PU_NAME_ON_REQACTPU`), use a plus sign (+) to add the `CLI_OVERWRITE_SYS_NAME` flag.

2. Issue either of the following commands, as needed:

```
- snaadmin -c define_node,ptf_flags=CLI_OVERWRITE_SYS_NAME
- snaadmin -c define_node,ptf_flags=SUPPRESS_PU_NAME_ON_REQACTPU+ \
  CLI_OVERWRITE_SYS_NAME
```

3. Optionally, edit the `/etc/sna/sna_node.cfg` configuration file:

- a. Stop the server (`sna stop`).
- b. Edit the configuration file `<define_node>` record.
- c. Restart the server (`sna start`).

4. On the Remote API Client, create a file with the name `tpnamehost` in the installation directory:

- For Windows clients, the installation directory is `C:\ibmcs\w64cli` or `C:\ibmcs\w32cli`.
- For Linux clients, the installation directory is `/etc/opt/ibm/sna`.
- For AIX clients, the installation directory is `/etc/sna`.

The content of the file does not matter, so it can be small:

```
echo > c:\ibmcs\w64cli\tpnamehost
```

Registering the DNS name of the client to the LU on the server

The client registers the DNS name for an LU when it initiates a connection through the server to the partner LU using the Local LU alias for the Local LU. Using `APING` on the client, for example, will set the `sys_name` parameter of the Local LU on the server to the DNS name of the client.

Figure 4-2 is an example where the client DNS name, `dicsw4.rtp.raleigh.ibm.com`, is set for the LU `DICSR2X`, with LU alias `dicsr2x`. Note that only the first 10 characters of the Machine, or `sys_name`, is displayed on the `snaadmin status_lu62` output.

The screenshot shows a Windows 64-bit client environment. The top window is a Command Prompt with the following output:

```

C:\IBMCS\w64cli>set APPCLLU
APPCLLU=dicsr2x

C:\IBMCS\w64cli>ping -n #batch -i 1 usibmmn.csa51in
IBM aping version 2.44 APPC echo test with timings.
Licensed Materials - Property of IBM
(C) Copyright 1994,1995 by IBM Corp. All rights reserved.

Allocate duration:      160 ms
Program startup and Confirm duration:      0 ms

Connected to a partner running on: CS/AIX 6.4
Duration      Data Sent      Data Rate      Data Rate
(msec)        (bytes)         (KB/s)         (Mb/s)
-----
Totals:       20           200            9.8            0.078
Duration statistics: Min = 20 Ave = 20 Max = 20
C:\IBMCS\w64cli>
  
```

The bottom window is a web browser displaying the IBM Communications Server for Linux Web Administration interface. The URL is `http://dicsr2.rtp.raleigh.ibm.com/cscbin/cs_callsnaadmin.cgi?query=status_lu62`. The page shows the following table of LU status:

LU	LU alias	Machine	Partner LU	Mode	Session Count
DEPLU02	deplu02				Inactive
DEPLU03	deplu03				Inactive
DEPLU04	deplu04				Inactive
DICSR2	dicsr2		USIBMNM.NM01M	CPSVRMGR	2 Sessions
			USIBMNM.LTLWGNN	CPSVCMG	2 Sessions
DICSR2T	TPLU2				Inactive
DICSR2X	dicsr2x	dicsw4.rtp	USIBMNM.CSA51IN	#BATCH	1 Session
DISCR2Y	discr2y				Inactive
I0BBD500	I0BBD500				Inactive

A red arrow points from the 'Machine' column of the table to the command prompt output, highlighting the client DNS name `dicsw4.rtp`.

Figure 4-2 Client DNS name setting

4.5 Remote API Client and IBM Personal Communications

On a Windows 64-bit client, no SNA stack is available for IBM Personal Communications. However, the Remote API Client and Personal Communications can be merged, so you still will be able to start SNA sessions in Personal Communications on 64-bit clients.

To merge Remote API Client and Personal Communications, follow this installation sequence:

1. Install the Remote API Client.
2. Run this script:

```
@ECHO OFF
REM
REM Set trace CLIENT_TYPE and COMMPKG in the IBM NSCMPKG support
REM
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\NSCMPKG\CLIENT_TYPE" /v
CLIENT_TYPE /f /d "CS/Linux"
REM
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\IBM\NSCMPKG\COMMPKG" /v
COMMPKG /f /d "API_client"
REM
```

3. Install Personal Communications V6.

The installation process for Personal Communications will detect the Remote API Client SNA stack, and then install the appropriate libraries to enable configuration of SNA resources.

4. Now, you will be able to configure LU6.2 and LU 0-3 sessions in Personal Communications, as shown in Figure 4-3.

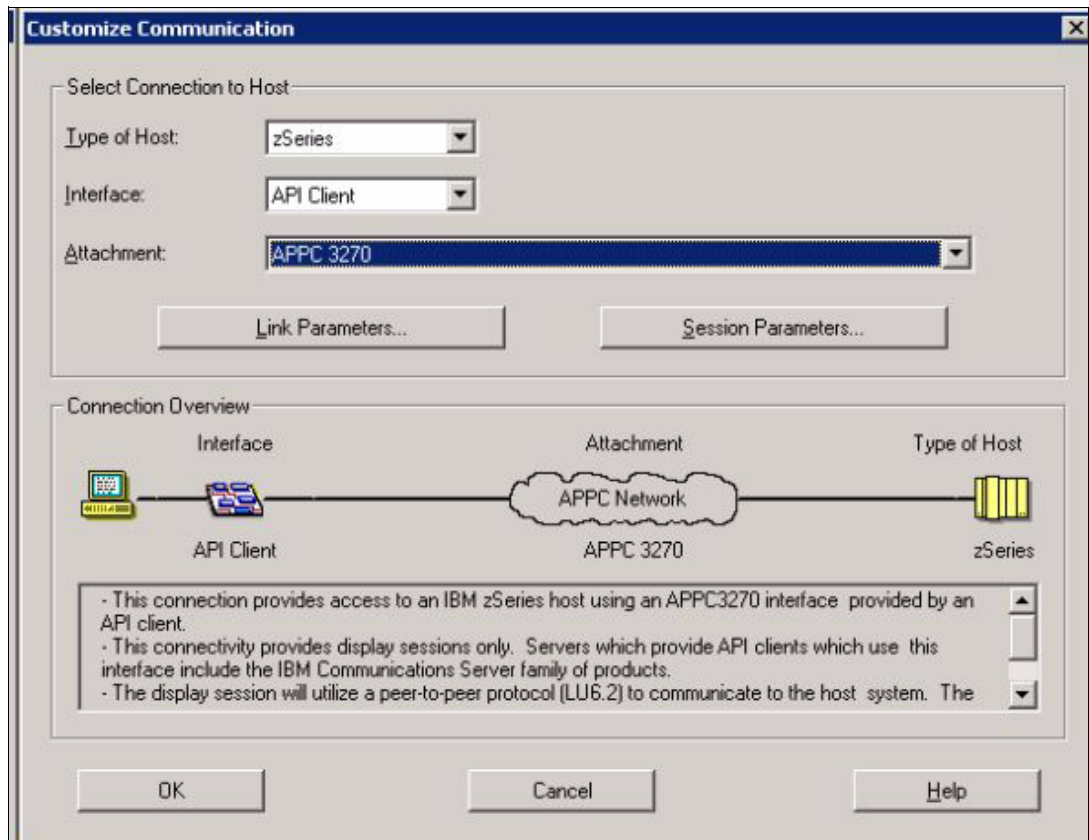


Figure 4-3 Configure LU6.2 and LU 0-3 sessions



Web Administration package

The Communications Server for Linux (CS Linux) Web Administration (WebAdmin) package is a tool designed to provide remote web connectivity for managing a CS Linux server installation. A full range of remote query and status information is provided, in both general and specific functional areas. Some start and stop functions for node, port, connection, and host resources are also provided.

In addition, some useful diagnostic tools for examining logs, taking traces, and capturing problem determination data are provided. Use these tools for most normal operations, but not for large query operations. If you need to query results for more than several hundred logical units (LUs) or connections, use the `snaadmin` command-line tool through direct Telnet (TN) or Secure Shell (SSH) connections.

The package depends on Common Gateway Interface (CGI) capabilities included with the Perl package for Linux. You must have an Apache server or IBM HTTP Server (IHS) and the Perl-CGI Red Hat Package Managers (RPMs) installed. The distributions tested for this package are those supported for the CS Linux products.

You can download the WebAdmin package from the following web page:

<http://www.ibm.com/support/docview.wss?uid=swg24008320>

This package provides the scripts and instructions for enabling remote web browser access to administrative functions for IBM Communications Server for Data Center Deployment.

Using a web server, such as Apache or IHS, this package provides remote access to the administrative functions, such as query, status, start, and stop, which you normally find on the Motif-based `xsnaadmin` GUI. This package is designed to provide the administrator with a secure and safe method for remote administration of the CS Linux resources. The same package will run on Intel, System p, and System z platforms that are supported by CS Linux.

This package uses Perl-CGI and Perl scripts to execute the administrative functions. Almost all web browsers work well with this package. The commands that are run to query and manage resources are found in the `snaadmin` command-line tool (see the section on the Command-Line Administration program in the `readme` file of the WebAdmin package).

The WebAdmin package provides access to resources controlled in the following panels:

- ▶ Node and Domain
- ▶ Connections
- ▶ Host Resources
- ▶ TN3270E Server
- ▶ Advanced Program-to-Program Communication (APPC)
- ▶ Advanced Peer-to-Peer Networking (APPN)
- ▶ Query
- ▶ Status
- ▶ Diagnostic

With these web interfaces, you can query any resource on the node, and get the status of sessions, links, LUs, and node conditions.

The web pages provide the interfaces to display link, physical unit (PU), and node conditions. You can start or stop the node, links, and PUs as needed. Host connectivity resources can be displayed in the same manner as the command-line administration calls, with additional, concise displays for management purposes. Enhanced displays for links, pools of LUs, PUs, and TN3270E server sessions are available.

Remote access is protected by user ID and password authentication, using Linux-supplied password-protection packages.

There are several package dependencies:

- ▶ A web server, normally the Apache server included with Linux distributions supported by the Communications Server for Linux, is required to use the WebAdmin package.
- ▶ The package was tested with the web servers included on the following Linux distributions:
 - Red Hat Enterprise Linux 5 and 6
 - SuSE Linux Enterprise Server 10 and 11
- ▶ Most web browsers that support CGI-enabled forms will work.
- ▶ For security purposes, you must also install the **sudo** package. This provides specific access to CS Linux Administration commands for specific users.

5.1 Installing WebAdmin

Follow these steps to install the WebAdmin package:

1. Download the CS Linux WebAdmin package from the following web page:

<http://www.ibm.com/support/docview.wss?uid=swg24008320>

2. Log in as root.
3. Install the WebAdmin package by using the following RPM command:

```
rpm -U ibm-commsserver-webadmin-version.rpm
```

The `ibm-commsserver-webadmin-version.rpm` file installs into the `/opt/ibm/sna/web` directory. This directory contains this readme file, and the following three subdirectories that the web server should reference (see Example 5-1).

Example 5-1 The content of the /opt/ibm/sna/web/ directory

```
/opt/ibm/sna/web/  
/opt/ibm/sna/web/README.html <- README file for package  
/opt/ibm/sna/web/sna.conf      <- HTTP configuration file aliases and  
package directory  
/opt/ibm/sna/web/rename_user.pl <- Script to change 'snauser' ID to another id  
in package  
/opt/ibm/sna/web/cscdoc       <- Directory for HTML pages  
/opt/ibm/sna/web/cscbin      <- Directory for scripts, CGI files used  
to perform requests  
/opt/ibm/sna/web/cscicons    <- Directory for icons in web pages
```

These files will all have `bin` user and `sna` group access rights.

5.2 Configuring WebAdmin

The WebAdmin package needs to be configured using the following steps:

1. Copy the `sna.conf` file from `/opt/ibm/sna/web` to the `conf.d` directory of your web server installation. This can be `/etc/httpd/conf.d` or `/etc/apache2/conf.d`. This will configure the `/cscdoc`, `/cscbin`, and `/cscicon` directories used to access the WebAdmin pages.
2. Use one of the following commands to set the user ID and password for access to the CS Linux WebAdmin pages:

```
- htpasswd -bc /opt/ibm/sna/.webpasswd admin passw0rd
```

```
- htpasswd2 -bc /opt/ibm/sna/.webpasswd admin passw0rd in Apache V2
```

This command creates `/opt/ibm/sna/.webpasswd` with an `admin` user and a password of `passw0rd`. Any reference to `http://<server>/cscdoc/` will prompt an initial access to request the proper user ID and password.

3. Set access for web server requests. Using root authority, run the following command:

```
/usr/sbin/useradd -g sna snauser
```

This creates a user called `snauser` in the `sna` group.

4. Run `ps -elf | grep http`, and make a note of the process owner for the httpd process. In this example, wwwrun is the process owner:

```
5 S wwwrun  31935  7862  0  80  0 - 1668 -      09:39 ?      00:00:00
   /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
```

5. Run `visudo` and add the following statement to the bottom of the `/etc/sudoers` file, which will open automatically:

Note: This should be entered as a single line.

```
<owner> <interface> = (snauser) NOPASSWD:
/opt/ibm/sna/bin/snaadmin,/opt/ibm/sna/bin/snagetpd, /opt/ibm/sna/bin/sna
```

In this statement, `<owner>` is the process owner that you noted in step 4, and `<interface>` is the Linux host name (as reported using the `hostname` command). In addition, `(snauser)` is the user ID that was created in step 3 (the parentheses are required).

Use the `NOPASSWD` option so that the system will not prompt for the root password when the web server runs the script.

Using this example, the statement has the following syntax:

```
wwwrun fury = (snauser) NOPASSWD:
/opt/ibm/sna/bin/snaadmin,/opt/ibm/sna/bin/snagetpd, /opt/ibm/sna/bin/sna
```

Note: Take care to verify that the paths are fully provided, to ensure that rogue execution will not be possible.

6. If pages are not properly invoked, view the `/var/log/httpd/error_logs` for messages.
7. Issue one of the following commands, as appropriate, to make these changes take effect on the web server:

- `/usr/sbin/apachectl restart`
- `/usr/sbin/apache2ctl restart`

Note: On some Hypertext Transfer Protocol (HTTP) servers, you will get the following warning messages:

- ▶ The alias directive in `/etc/apache2/conf.d/sna.conf` at line 22 will probably never match because it overlaps an earlier alias.
- ▶ The alias directive in `/etc/apache2/conf.d/sna.conf` at line 40 will probably never match because it overlaps an earlier alias.

These messages are only informational, and do not necessarily indicate a problem.

8. Optional: Change the Systems Network Architecture (SNA) user default:
 - a. Before changing the user ID, you should back up the `/opt/ibm/sna/web/cscbin` directory first. This directory contains the files where the `snauser` ID is defined in the WebAdmin scripts.
 - b. If the user ID, `snauser`, is not an ID that you want to use, you can rename the user ID in the WebAdmin files by running the `/opt/ibm/sna/web/rename_user.pl` script. This will prompt you for the new ID to use.
9. These instructions for configuring the web server are designed to have `http://<server_name>/cscdoc/cslinuxweb.html` as the main page for the WebAdmin package. Add a link to this main page somewhere on the web server for easier access.

Add the following HTML code to any of your web pages to provide a link to the WebAdmin start page:

```
<a ref="http://<hostname>/cscdoc/cslinuxweb.html" > <img SRC="/icons/link.gif"
ALT="" align=BOTTOM > web page for CS Linux WebAdmin < /a >
```

In this code example, *<hostname>* is the access name for the web server.

WebAdmin Start panel

Figure 5-1 shows the start panel for the WebAdmin package.

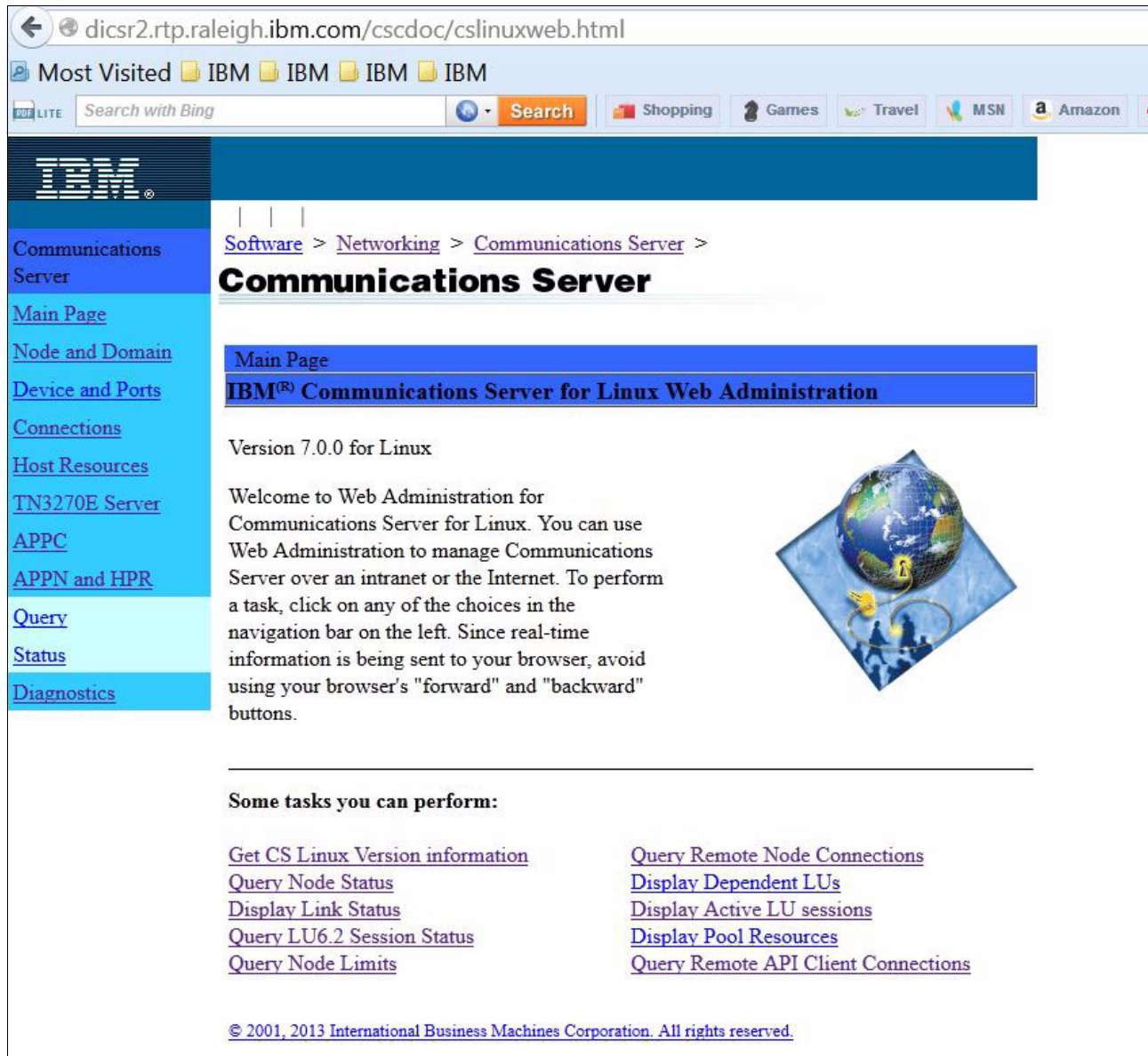


Figure 5-1 Start panel of the WebAdmin package

If the WebAdmin has problems displaying output, it can be due to one of these common problems:

- ▶ Permissions are not correct for running the `/opt/ibm/sna/bin/snaadmin` script:
 - Check that the line `# Defaults requiretty` is commented out in the `/etc/sudoers` file.
 - Check that the `snauser` ID is created and is in the `/etc/sudoers` statement for accessibility.
- ▶ Security Enhanced Linux (SELinux) is disabling the activation of `snaadmin` commands. See the following technote on getting SELinux to permit `snaadmin` commands:
<http://www.ibm.com/support/docview.wss?uid=swg21255980>

View the `/var/log/httpd/error_log` or `/var/log/apache2/error_log` file for web server error messages concerning access and execution of WebAdmin package scripts.

5.3 Administering WebAdmin

You can use the WebAdmin package to query SNA resources, and to view various status displays. There are some control panels provided that the administrator can use to start and stop links, traces, and the node.

5.3.1 Node management

The Communications Server has the following states for processing SNA traffic:

- ▶ The SNA software can be started using the `sna start` command, but the SNA node can still be in an `Inactive` state:
 - In this state, the process daemons and the configuration are in memory, but the SNA node is not actively communicating any SNA links, sessions, or messages.
 - Remote API clients can connect to the node, and resources that are active on other servers in the domain will be accessible. However, the control point for the local SNA node is not active.
- ▶ When the SNA node is activated, by the `snaadmin init_node` command, the control point performs the following functions:
 - Connects to other network nodes
 - Starts links that are configured to start on node activation
 - Allows control point-to-control point (CP-CP) sessions to be established

Figure 5-2 shows the Node and Domain panel. To navigate it and activate the node, follow these steps:

1. Select the **Node and Domain** link on the left of the panel. The Node and Domain panel also shows you which domain resources you can view.

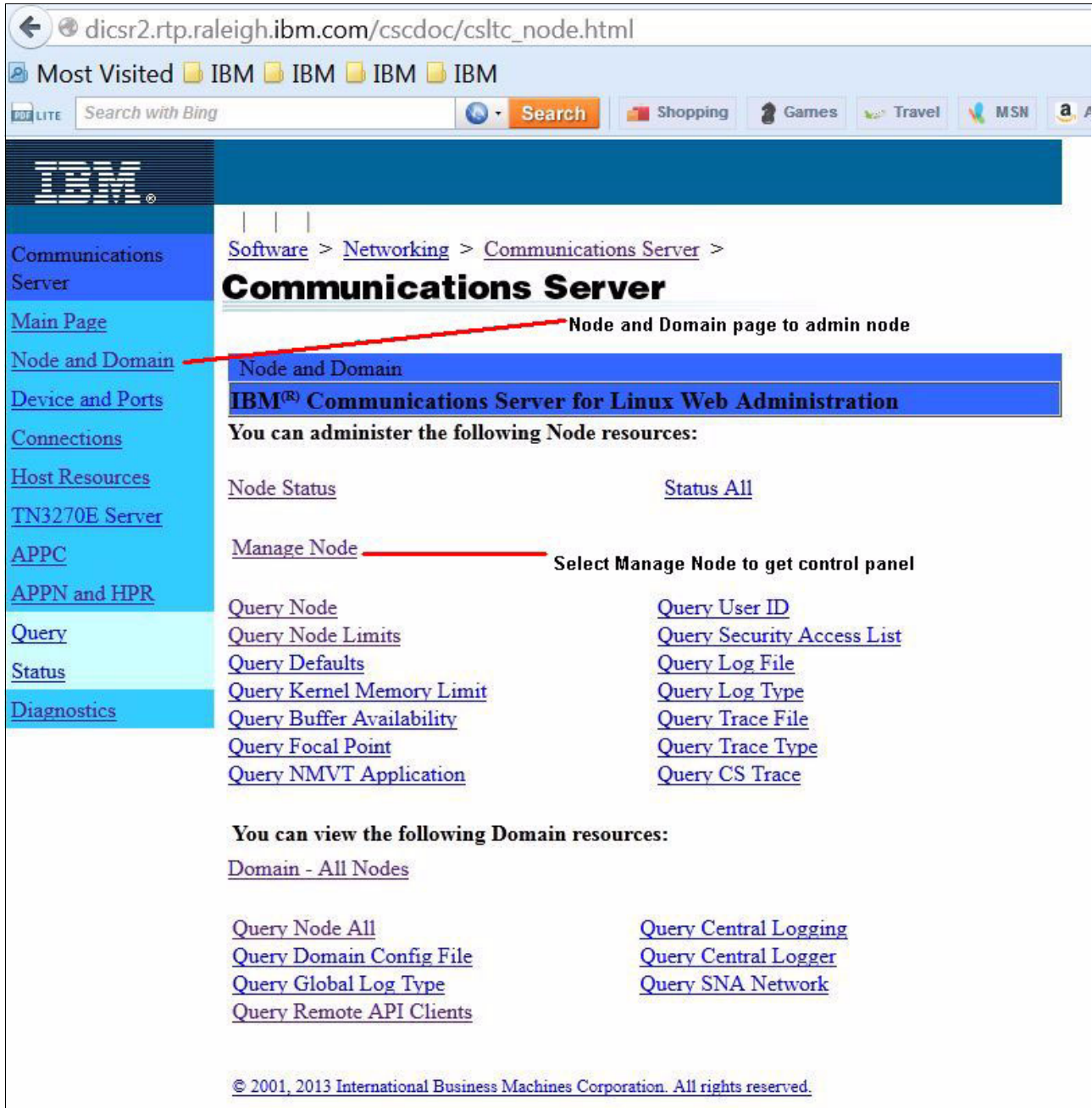


Figure 5-2 Communications Server Node and Domain window

2. Select the **Manage Node** link to display a control panel (as shown in Figure 5-3).
The display shows each node, its status (Active or Inactive), and its role (Backup or Master).
3. To start the node, select the **Start** check box and click **Submit**. You also have the option to stop the SNA kernel and all of the SNA processes.
When you click **Submit**, the response shows the new status of the server.

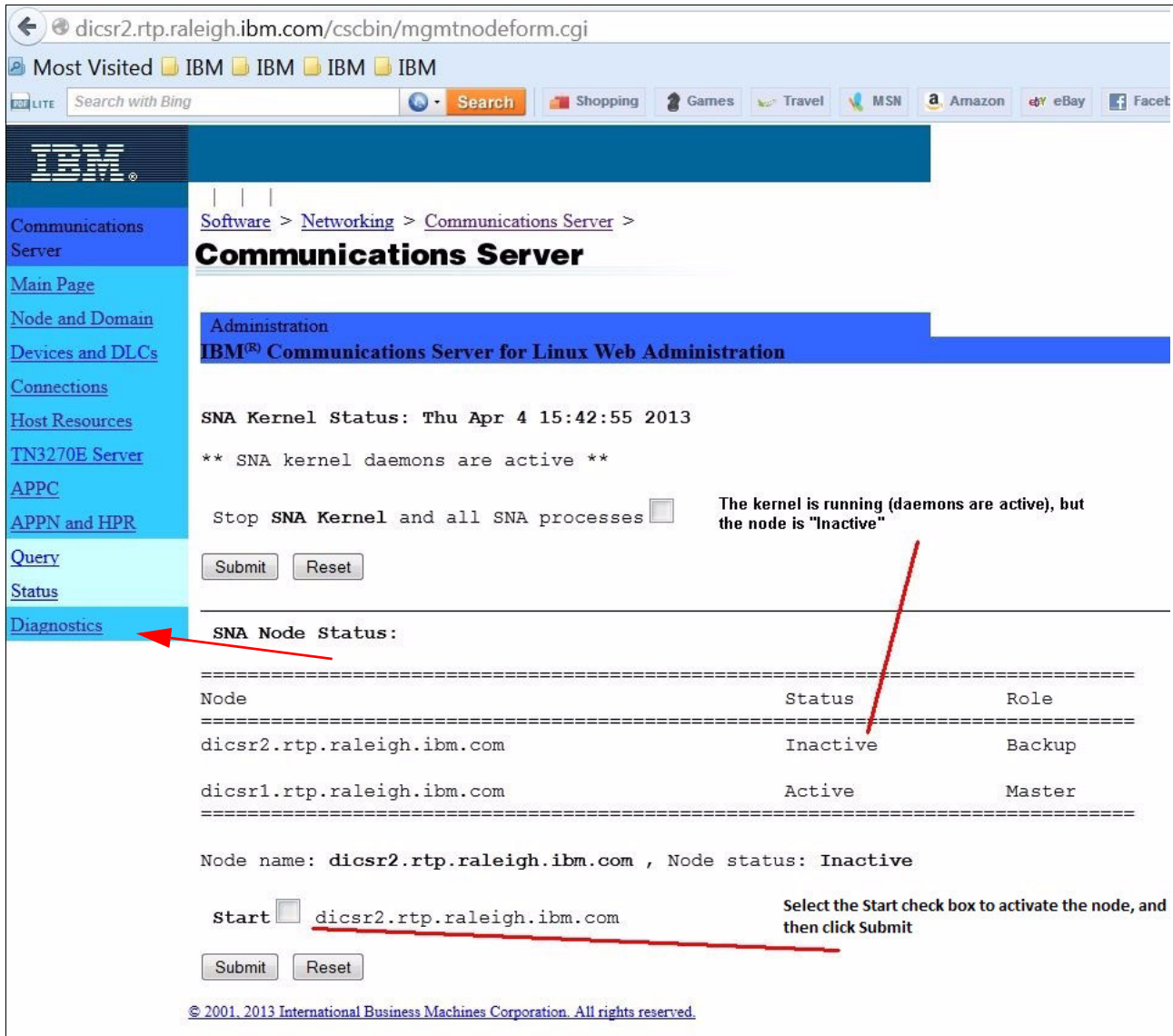


Figure 5-3 Communications Server status display

Diagnostics

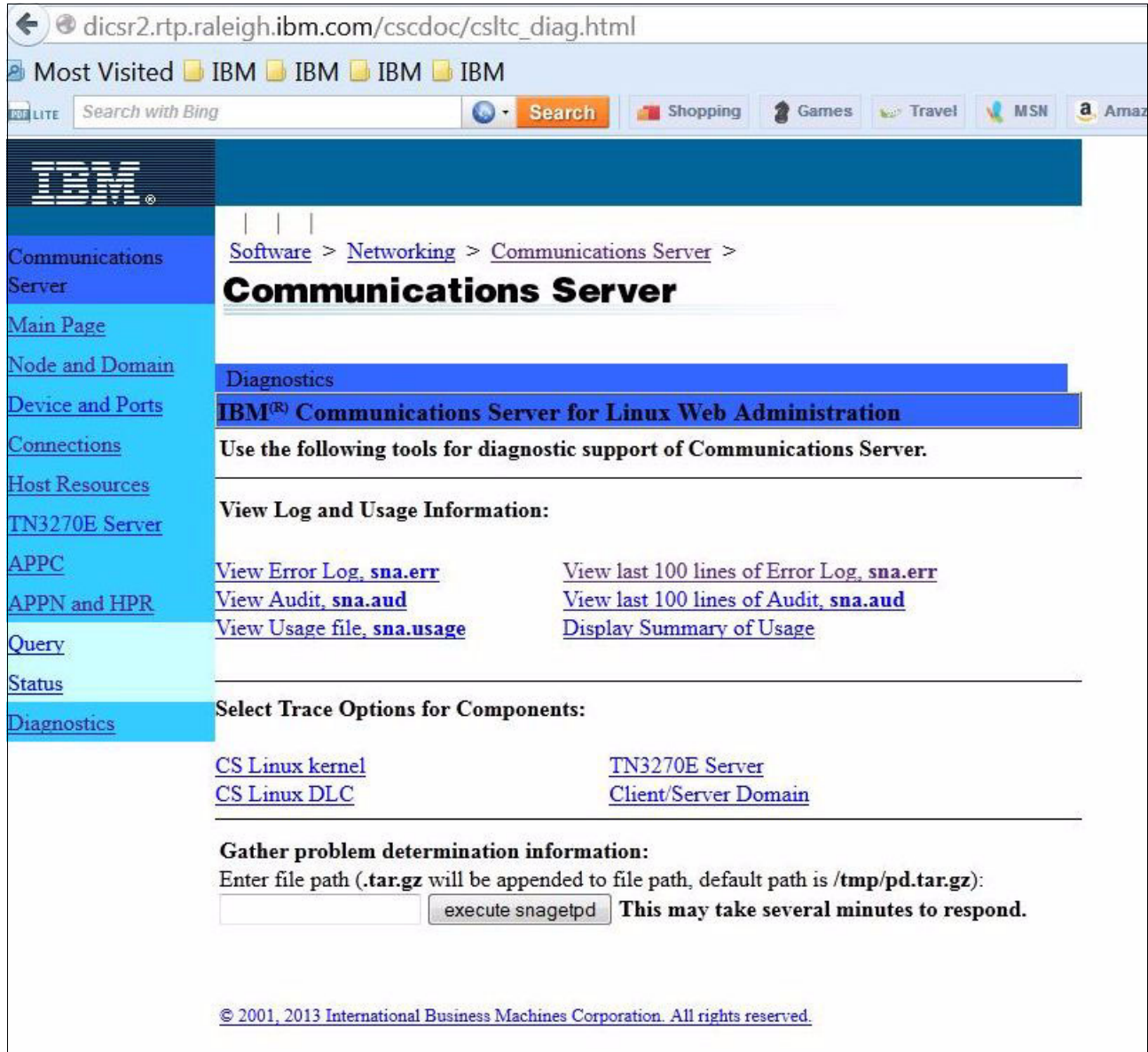
The Diagnostics panel is useful in examining logs and capturing traces through a web browser. The panel provides a view of the `sna.err` and `sna.aud` log files, or just the last 100 lines of the log files. The panel also provides links to panels for enabling the various trace options, and you can use it to capture a `snagetpd` problem determination file. This will stop traces, and gather the configuration files, logs, and system status for examination by support.

Note: It is important to notice the Diagnostics link in the left navigation panel.

Enabling tracing

Tracing can be enabled through the WebAdmin by going to the Diagnostics panel and selecting one of the trace options. From within the trace dialog panel, select the type of trace needed. The **All** option activates all traces. If any trace options are already active, a **None** option is provided to disable all of the trace options for that type of trace.

Figure 5-4 shows what trace options are available for client/server traces.



Software > Networking > Communications Server >

Communications Server

[Diagnostics](#)

IBM^(R) Communications Server for Linux Web Administration

Use the following tools for diagnostic support of Communications Server.

View Log and Usage Information:

View Error Log, <u>sna.err</u>	View last 100 lines of Error Log, <u>sna.err</u>
View Audit, <u>sna.aud</u>	View last 100 lines of Audit, <u>sna.aud</u>
View Usage file, <u>sna.usage</u>	Display Summary of Usage

Select Trace Options for Components:

CS Linux kernel	TN3270E Server
CS Linux DLC	Client/Server Domain

Gather problem determination information:
Enter file path (.tar.gz will be appended to file path, default path is /tmp/pd.tar.gz):
 This may take several minutes to respond.

© 2001, 2013 International Business Machines Corporation. All rights reserved.

Figure 5-4 Communications Server Diagnostics window

Related publications

The publications listed in this section are considered particularly suitable for gaining more detailed information about the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only:

- ▶ *SNA Modernization Strategy for Access Node Connectivity*, REDP-4871
<http://www.redbooks.ibm.com/abstracts/redp4871.html?Open>
- ▶ *IBM Communications Server for AIX, V6: New Features and Implementation Scenarios*, SG24-5947
<http://www.redbooks.ibm.com/abstracts/sg245947.html?Open>

Online resources

In this IBM Redpaper publication, we refer to the following documentation:

- ▶ *IBM Communications Server for Data Center Deployment on Linux, Quick Beginnings*
<ftp://public.dhe.ibm.com/software/network/commsserver/publications/csdcdv7/czx00110.pdf>
- ▶ *IBM Communications Server for Data Center Deployment on Linux on System z, Quick Beginnings*
<ftp://public.dhe.ibm.com/software/network/commsserver/publications/csdcdv7/czx00210.pdf>
- ▶ *IBM Communications Server for Data Center Deployment on AIX, Quick Beginnings*
<ftp://public.dhe.ibm.com/software/network/commsserver/publications/csdcdv7/czxdys10.pdf>
- ▶ *IBM Communications Server for Data Center Deployment on Linux, Administration Guide*
<ftp://public.dhe.ibm.com/software/network/commsserver/publications/csdcdv7/czx00410.pdf>
- ▶ *IBM Communications Server for Data Center Deployment on Linux, Administration Command Reference*
<ftp://public.dhe.ibm.com/software/network/commsserver/publications/csdcdv7/czx00310.pdf>
- ▶ *IBM Communications Server for Linux V6.2 APPC Application Suite User's Guide*
<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC31-6772-00>

- ▶ *IBM Communications Server for Linux V7.0: APPC Application Suite User's Guide*

<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC23-8595-01>

- ▶ *IBM Communications Server for Data Center Deployment on AIX Administration Guide Version 7.0*

<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss?CTY=US&FNC=SRX&PBL=SC31-8586-05>

The online documentation for Communications Server for Data Center Deployment is located on the library reference page:

<http://www.ibm.com/support/docview.wss?uid=swg27005371>

The WebAdmin package is located at the following URL:

<http://www-01.ibm.com/support/docview.wss?uid=swg24008320>

How to get Redbooks

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, drafts, and additional materials, at the following website:

<http://ibm.com/redbooks>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



IBM Communications Server for Data Center Deployment V7.0



Deployment on AIX, Linux, and Linux on System z

Ability to run SNA applications via IBM Remote API Client

New deployment features for the client/server implementation

This IBM Redpaper publication will guide the user through the installation, configuration, and administration of IBM Communications Server for Data Center Deployment V7.0. It is not intended to be all-inclusive. Rather, it builds on previous publications referenced throughout the document. The focus is on consolidation of Systems Network Architecture (SNA) resources, key features and functions available in IBM Communications Server for Data Center Deployment, and the Web Administration package specific to the Linux platform.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers, and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-4998-00